JULIO 2018

TRABAJO FIN DE GRADO

DESARROLLO DE UN SIMULADOR REGULADOR DEL TRÁFICO URBANO PARA WINDOWS

AUTOR:
Andrés Meseguer Valenzuela

TUTOR:
D. Carlos Domínguez Montagud

COTUTOR:
D. Houcine Hassan Mohamed

GRADO EN INGENIERIA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

Agradecimientos

A mi tutor y cotutor por toda la paciencia que han tenido conmigo. A mis padres y hermano por la educación que me han dado. A los amig@s que siempre están ahí. A mis familiares de Levante y Andalucía, y en especial a mi albuñanera. Siempre te recordaremos.

RESUMEN

En este proyecto se ha desarrollado una aplicación para Windows en la cual, se pueden llevar a cabo simulaciones del comportamiento del tráfico en Zonas Urbanas creadas por el usuario, haciendo posible el control automático de los ciclos de semáforos en función de valores predefinidos anteriormente además de almacenar y exportar los valores obtenidos para su posterior análisis.

RESUMEN

En aquest projecte s'ha desenvolupat una aplicació per a Windows amb la qual, es poden dur a terme simulacions del comportament del tràfic en Zones Urbanes creades per l'usuari, fent possible el control automàtic dels cicles dels semàfors en funció de valors redefinits anteriorment, a més d'emmagatzemar i exportar els valors obtinguts per analitzar-los posteriorment.

ABSTRACT

In this project a software for Windows has been developed in which simulations of traffic behaviour in urban areas can be carried out. These urban areas are created by the user making it possible to produce an automatic control of traffic light cycles based on values previously predefined, apart from to store and export the values obtained for further analysis.

Contenido

Introducción

Visión General

Los atascos, los problemas de calidad del aire, las enfermedades que afectan al sistema respiratorio son inconvenientes que nos afectan en nuestro día a día como ciudadanos de la época actual.

No quedando ahí, estos problemas que afectan a nuestra salud y vida profesional acaban creando más problemas con el resultado de una menor calidad de vida.

Por todo ello, la temática de este trabajo tiene en cuenta las causas del problema. Desde que encendemos el motor del coche hasta los momentos en que pisamos el freno. Desde que salimos de nuestros hogares en Zonas Residenciales hasta cuando nos desesperamos durante atascos en Zonas Urbanas de alta densidad.

Planteando una herramienta que afecta a las causas principales desde que el semáforo se pone en Rojo, hasta que se pone en Verde, y más concretamente, en el tiempo que tarda en llevarse a cabo todo este proceso.

Estado de la Cuestión

Para poder llevar a cabo el acercamiento al tema de estudio, es necesario un primer punto de aproximación a este trabajo. Este se encuentra en una denuncia de la ONG Ecologistas en Acción que expresa los siguientes puntos:

- Las ciudades de Barcelona, Guadalajara, Salamanca, Sevilla, Zaragoza y Madrid, han superado el nivel máximo de dióxido de nitrógeno establecido por normativa (200 microgramos por metro cúbico).
- Un total de 16 ciudades han superado el límite del contaminante PM10, el cual es más peligroso que el Dióxido de Nitrógeno.
- La contaminación del Aire en España provoca más de 30.000 muertes anuales, según la OMS (Organización Mundial de la Salud).

Conocido el contenido de esta denuncia, es posible entender que la Contaminación del Aire en España es un problema de alcance debido a la gran cantidad de ciudades a las que afecta a parte de ser una causa de mortalidad importante.

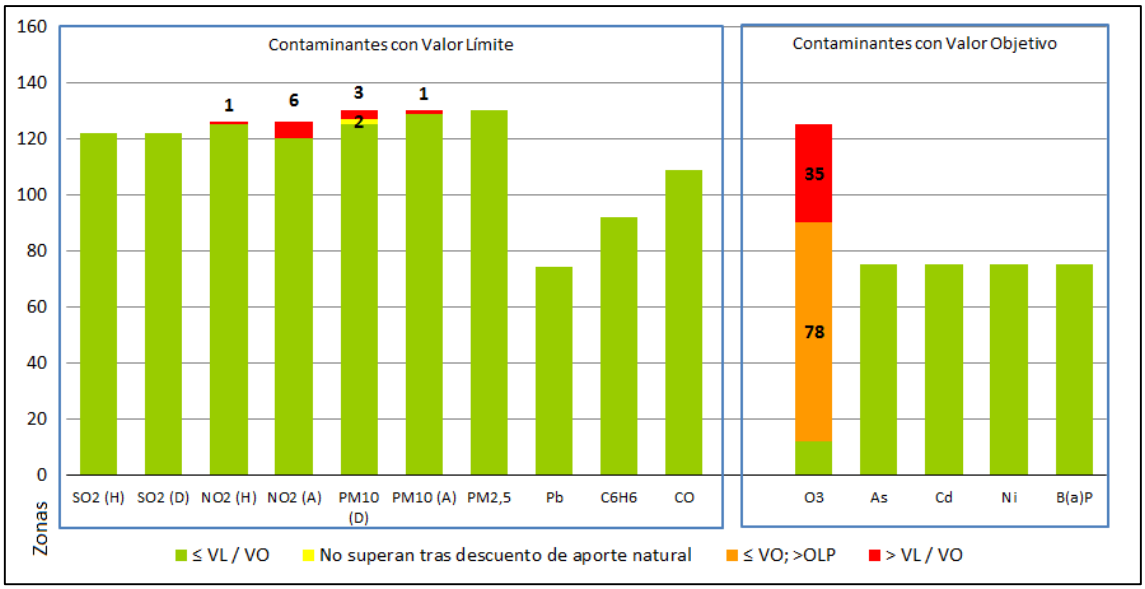
Por todo ello, se decidió buscar estudios que especificaran los contaminantes principales del Aire, las principales fuentes de estos y las consecuencias de este tipo de contaminación.

Los principales estudios que se han tenido en cuenta junto con sus principales ideas son los siguientes:

Informe de Evaluación sobre la Calidad del Aire 2016

- Realizado por el Ministerio de Agricultura y Pesca, Alimentación y Medio Ambiente.
- La Evaluación se llevó a cabo para los siguientes contaminantes: Dióxido de Azufre (SO2), Dióxido de Nitrógeno (NO2), Óxidos de Nitrógeno (NOx), Partículas (PM10 y PM2,5), Plomo (Pb), Benceno (C6H6), monóxido de carbono (CO), Ozono (O3), Arsénico (As), Cadmio (Cd), Níquel (Ni) y benzo(a)pireno (B(a)P).
- Los resultados del estudio defienden que de los principales contaminantes únicamente el **dióxido de azufre (SO2), el material particulado (PM10)** y el **ozono troposférico (O3)** han superado los valores límite estipulados.

Ilustración 1.0:Resumen de las superaciones en 2016 por contaminante.



Nota: Recuperado del Informe de Evaluación sobre la Calidad del Aire 2016, Ministerio de Medio Ambiente.

La información del informe del Ministerio de Medio Ambiente ofrece unas conclusiones que se contradicen en parte con el segundo estudio que se ha tenido en cuenta:

La calidad del aire en el Estado español durante 2016

- Realizado por Ecologistas en Acción.
- El estudio analiza la calidad del aire que respira la población española (46,6 millones de personas), en relación a la protección de la salud humana.
- Los contaminantes más problemáticos en el Estado español durante 2016 han sido las Partículas en Suspensión (PM10 y PM2.5), el Dióxido de Nitrógeno (NO2), el Ozono Troposférico (O3) y el Dióxido de Azufre (SO2).
- La población que respira aire contaminado en el Estado español, alcanza los 16,9 millones de personas, es decir un 36,4% de toda la población. En otras palabras, uno de cada tres españoles respira un aire que incumple los estándares legales vigentes.
- Si se tienen en cuenta los valores recomendados por la Organización Mundial de la Salud (OMS), la población que respira aire contaminado se incrementa hasta los 43,7 millones de personas. Es decir, un 93.9% de la población.
- Reducción de los principales contaminantes debida más a razones coyunturales (la crisis económica) que a la aplicación de medidas planificadas.
- Tras cuatro décadas de regulación legal, los contaminantes clásicos (partículas, NO2 y SO2) siguen afectando a casi tres cuartas partes de la población española.
- La contaminación del aire es un asunto muy grave, que causa más de 24.000 muertes prematuras en el Estado español cada año.
- La Comisión Europea inició en enero de 2009 un procedimiento de infracción contra España por el incumplimiento de la normativa sobre calidad del aire respecto a las partículas PM10.

Conocidos estos estudios y sus conclusiones más destacables, es importante remarcar ante todo la diferencia de límites sobre la concentración de Contaminantes en el Aire. Diferenciando los límites estipulados por la propia administración respecto a los de la OMS (Organización Mundial de la Salud), los cuales son más exigentes.

Además, resaltar que el estudio de Ecologistas en Acción lleva a cabo una crítica más exigente al tener en cuenta los principales límites establecidos, llegando a la conclusión que, según los valores de la OMS, el 93.9% de la población respira aire con niveles superiores a los límites de la organización.

A parte de estos estudios, existen infinidad de artículos de prensa, análisis, informes y otros estudios que alertan de que el tráfico rodado es una de las principales fuentes de contaminación atmosférica y acústica, denunciando que se está reduciendo la esperanza de vida y afectando al desarrollo de los niños, definiendo así los límites de este problema.

○ Justificación

Conocidos los datos, nos es posible argumentar que debido a la gran cantidad de afectados por la Contaminación del Aire y al tráfico rodado como uno de las fuentes de este problema, la creación de una Simulador que permita mejorar la eficiencia de los sistemas de tránsito en Zonas Urbanas queda justificada.

○ Alcance y Límites

Debido a las limitaciones temporales y económicas, el proyecto a realizar tendrá en cuenta un número muy delimitado de variables que deben ser obtenidas experimentalmente mediante un estudio para que los resultados se ajusten a la realidad.

Además, argumentar la limitación del proyecto dado que la comprobación del funcionamiento solo es posible únicamente mediante el propio software, dada la dificultad de maquetación del tránsito rodado.

○ Capítulos.

El proyecto justificado se va a explicar siguiendo el siguiente orden:

- **Objetivos**: Definición de las metas que se pretender lograr.
- **Desarrollo**: Explicación del planteamiento teórico establecido además de los pasos que han sido necesarios para su creación y los elementos empleados.
- **Pruebas Realizadas**: Seguimiento de los experimentos realizados.
- **Manual de Usuario**: Explicación general del uso de la aplicación.
- **Presupuesto**: Detalle de los costes que puede causar el desarrollo de este proyecto.
- **Conclusión**: Valoración de los resultados obtenidos. Ubicación del proyecto en su campo.

Objetivos

Llevar a cabo la realización de un simulador que permita señalizar y controlar zonas de transito urbanas en función de las siguientes variables que serán introducidas por el usuario:

- Cantidad de vehículos en función del tiempo
- Cantidad de peatones en función del tiempo
- Nº de carriles para cada zona señalizada
- Velocidades Medias.
- Tipos de semáforos.
- Posiciones de Inicio y Fin de las distintas zonas.

De modo que con estos valores se consiga ejercer el control sobre las distintas zonas dispuestas por el usuario y obtener valores a representar posteriormente, haciendo posible la obtención de tiempos aconsejables para cada semáforo que comporten una mejora en la fluidez y por tanto en la concentración de la contaminación anteriormente explicada.

Desarrollo

Planteamiento Teórico

Llevar a cabo el control de un semáforo es una tarea relativamente sencilla dado que predeterminadamente, su control se efectúa mediante temporizadores.

En cambio, con el paso del tiempo, algunas ciudades han empezado a añadir controles en función de la densidad del tráfico (vehículos en función del tiempo), lo que supone una tarea más ardua.

Dado que este problema se complica en función del tamaño del área de estudio porque a cuanta más área, mayor cantidad de semáforos, vehículos y peatones, ha sido necesario tener claro el planteamiento teórico para resolver el problema.

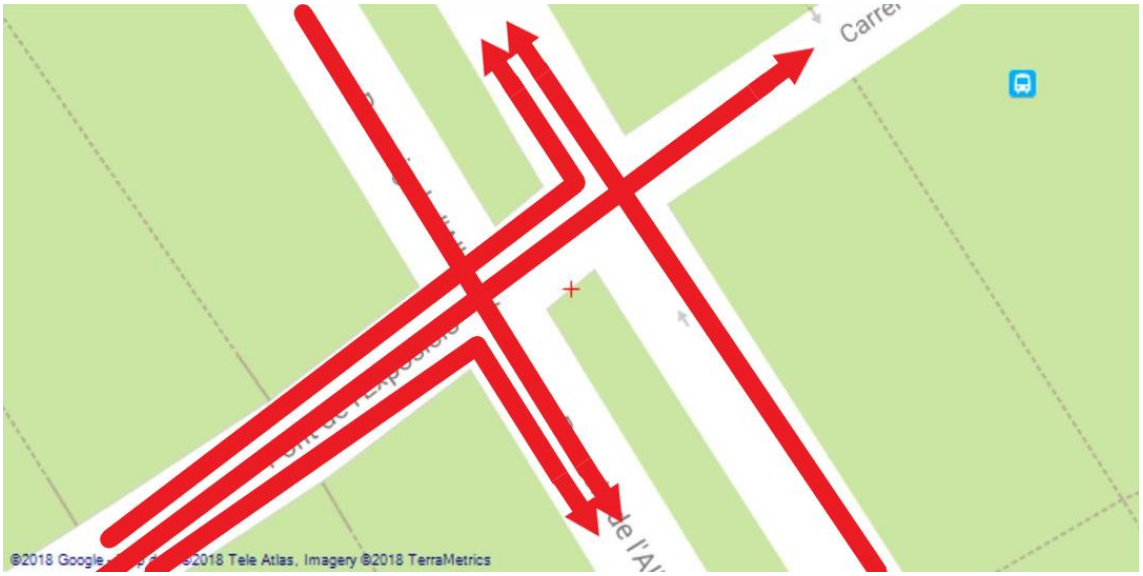
Es decir, si se dispone de una zona de tránsito que forme un cruce, esta tiene áreas de tránsito comunes por lo que debe ser regulada mediante semáforos, de modo que se asegure la alternancia de acceso desde las diferentes vías que conforman el cruce.

Un ejemplo como el propuesto arriba se puede observar en la ciudad de Valencia en el cruce formado por el Puente de la Exposición, el Paseo de la Alameda y la calle de Armando Palacio Valdés. (Coordenadas: 39°28'25.3"N 0°21'53.5"O).

La existencia de estas áreas comunes junto con la alternancia provoca que los vehículos, independientemente de la vía de donde procedan, tengan puntos intermedios en sus trayectorias que puedan ser compartidos con vehículos de otras vías con trayectorias distintas.

En la siguiente imagen se pueden observar las distintas trayectorias posibles en rojo.

Ilustración 2: Trayectorias



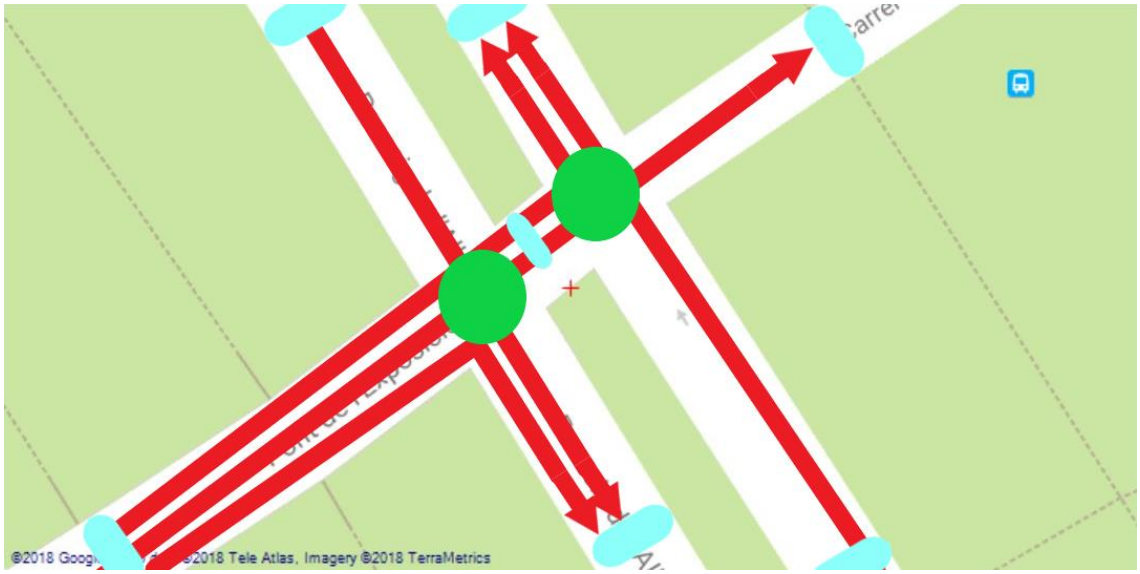
Basándose en estos puntos intermedios se extrajo el primer planteamiento teórico, en el cual primeramente se disponen las entradas y las salidas, es decir, los puntos donde se reciben vehículos con sentido hacia el interior de la zona de estudio y los puntos donde los vehículos transitan para abandonar esta área de estudio.

Una vez dispuestas las entradas y las salidas, se disponen los puntos intermedios, de forma que a estos se les relaciona con sus respectivas entradas y salidas. De este modo, un punto intermedio puede tener como **mínimo** una entrada y una salida.

Además, estos puntos intermedios reciben los valores de interés o variables que se explican posteriormente, provocando que los puntos intermedios contengan toda la información necesaria para poder decidir que entrada y que salida activar.

En la siguiente imagen se puede observar la estructura anteriormente explicada, compuesta por las trayectorias en rojo, los puntos de entrada y salida en azul además de los puntos intermedios en verde.

Ilustración 3: Disposición mediante puntos Intermedios.



El hecho de disponer de zonas de estudio con un punto intermedio en cada una, de modo que estos puntos conozcan todas las trayectorias posibles y tengan todos los datos para decidir, proporciona seguridad pero no fluidez, dado que, en estos casos, solo habrá una trayectoria activa, es decir, una entrada y una salida conectadas a la vez.

Para aumentar la fluidez es necesario aumentar el área de la zona de estudio y distribuir una mayor cantidad de puntos intermedios, entradas y salidas con lo cual se pueden distinguir los principales problemas de este planteamiento:

- Necesidad de relacionar los puntos intermedios entre sí, con la dificultad de relacionar que la entrada para un punto intermedio puede suponer una salida para otro punto intermedio.
- Dificultad creciente en zonas con diferentes salidas simultaneas posibles (común en rotondas).

Ante todos estos problemas surgió la necesidad de simplificar el planteamiento hasta el punto de reducir las zonas de estudio al mínimo posible para estudiarlas individualmente, de modo que cada una de estas zonas de estudio tengan un comportamiento determinado que afecta a las zonas vecinas, llevando a cabo un planteamiento de inteligencia de enjambre, de modo que no exista una estructura de control centralizado sino que cada zona mínima tenga su propio control pero siempre

conectada con otras zonas mínimas, de modo que el comportamiento de una afecte a las demás.

El nombre adquirido en el estudio a estas nuevas zonas mínimas es el de **Tramos**.

Los Tramos son cada parte de la vía publica dedicada al tránsito de vehículos que comienzan en un punto I y terminan en un punto F (donde pueda existir un semáforo o sencillamente finalice en este punto), entre los cuales se transita en el mismo sentido, independientemente del número de carriles.

Con el Tramo y su definición, se da pie a un nuevo planteamiento en el cual, cada Tramo ejerce de una parte de la vía pública a la que se aportan unos datos con los que decidir su estado.

Dado que los Tramos representan partes pequeñas de las trayectorias del planteamiento con puntos intermedios, esta situación provoca que se puedan resolver situaciones delicadas con múltiples salidas, además de hacer más sencilla, la relación entre Tramos respecto a la relación entre zonas con puntos intermedios, dado que en el primer caso los Tramos no comparten áreas, en cambio, las zonas con puntos intermedios pueden compartir entradas y salidas.

En resumen, el hecho de reducir la zona de análisis ha provocado una mayor simplificación para poder controlar una zona más grande.

	Planteamiento por Tramos	Planteamiento por Puntos Intermedios
Ventajas	- Control Descentralizado. - Relación simple entre zonas.	Sencillo para zonas con pocas trayectorias y de un solo punto intermedio.
Desventajas	Mayor cantidad de zonas individuales sobre las que operar.	Difícil relación entre puntos intermedios.

Tabla 1: Ventajas y Desventajas de los Planteamientos.

○ Variables.

Una vez conocidos los dos planteamientos surgidos a lo largo del desarrollo del proyecto, quedando claro que el planteamiento por Tramo es el escogido para llevar a cabo el control durante la simulación, es necesario conocer las diferentes variables con las que se trabaja.

Las variables a estudiar para cada tramo son las siguientes:

- **Estado**: Determina si el semáforo ubicado en el punto Final está en verde o en rojo, indicando por lo tanto, si los vehículos pueden abandonar o no el Tramo. Aquellas áreas de estudio sin semáforo, se les tiene en cuenta que siempre permiten el tránsito.
- **Carriles**: Nº de carriles que componen los Tramos, su valor ayuda a poder calcular la capacidad del Tramo (nº máximo de vehículos que pueden ocuparlo).
- **Coordenadas del Punto Inicial y Final**: Latitud y longitud de los dos puntos que delimitan el área de estudio. Las coordenadas serán valores decimales tanto para latitudes como longitudes.
- **Tipo de Semáforo**: Valor que depende del semáforo del Tramo, el cual siempre está ubicado en el punto Final. Puede surgir que el área de estudio no tenga ningún semáforo, por lo tanto, esta variable tendrá valor 0.
- **Vehículos**: Cantidad de vehículos que se encuentran circulando por los Tramos.
- **Peatones**: Cantidad de peatones que esperan la pausa de la circulación en cada Tramo.
- **Importancias**: Porcentajes que relacionan la cantidad de los vehículos de un determinado Tramo que tienen intención de dirigirse al Tramo de estudio.
- **Distancia**: Distancia desde el punto inicial hasta el punto final de los Tramos, por lo tanto, su valor depende directamente de las coordenadas de estos puntos.
- **Velocidad Media**: Valor en Km/h de la celeridad media de los vehículos que circulan por el Tramo.
- **Capacidad**: Cantidad de vehículos que pueden ocupar un Tramo. Valor directamente relacionado con la distancia y el nº de carriles.

- **Variable Vehículos**

De todas ellas, la variable Vehículos es el valor a controlar durante la simulación, de modo que todas las demás van a tenerse en cuenta para poder llevar a cabo la regulación de la cantidad de vehículos en todos los Tramos. Esta cantidad se calcula para cada ciclo (i), mediante la siguiente expresión:

$$Vehículos_i = Vehículos_{entrantes} + Vehículos_{i-1} - Vehículos_{salientes}$$

Con la expresión anterior se entiende que la cantidad de vehículos depende en sí de los Vehículos entrantes y salientes además de los que permanecían en el periodo anterior que inicialmente ($i=0$) serán nulos.

Dado que el valor de los vehículos en el periodo anterior es conocido, el control de la variable Vehículos se basa en obtener la cantidad de transportes entrantes y salientes, los cuales están relacionados con las demás variables.

Vehículos Entrantes

Empezando por las Importancias, si el Tramo de estudio no tiene ningún valor de Importancia en ningún otro Tramo, nuestra área de estudio se le considera como Entrada Principal, debido a que no recibe vehículos de ningún otro Tramo, sino que los recibe de vías públicas que no son objetivo de simulación o control.

Por lo tanto, a todo Tramo sin importancias se le denomina como Entrada Principal. En cambio, todo Tramo debe tener a la salida un sumatorio de importancias del 100%.

A los Tramos considerados Entradas Principales se les atribuye un flujo de vehículos, es decir, una determinada cantidad de vehículos en función del tiempo. Este flujo se modeliza como un tipo de función determinado con sus valores característicos, como lo puede ser una distribución normal con media (μ) y desviación (σ).

En caso de no tratarse de Entrada Principal, el Tramo recibe una cantidad de vehículos de cada Tramo en función de las Importancias.

Como se trata de un programa de ordenador, es decir, de un sistema discreto y que por lo tanto, funciona periódicamente, únicamente se van a tener en cuenta los vehículos que abandonen su Tramo en el mismo período.

Por lo tanto, si se está estudiando un Tramo X, el cual tiene relacionados a su entrada por Importancias los Tramos 1, 2 y 3, en el primer paso se tienen en cuenta cuales son estas importancias relacionadas:

- El Tramo 1 tiene 2 (n_1) vehículos de los cuáles el 50% (I_1) se dirigen a X.
- El Tramo 2 contiene 3 vehículos (n_2) de los cuales el 100% (I_2) tienen como destino X.
- El Tramo 3, tiene 2 vehículos (n_3) con un 50% de Importancia (I_3) para el Tramo X sobre la salida de este Tramo.

El segundo paso es conocer los estados de los Tramos:

- E1: Activo (*su semáforo está en verde, ámbar o sencillamente es nulo*).
- E2: Parado (*su semáforo está en rojo*).
- E3: Parado (*su semáforo está en rojo*).

Conocidos los valores de Importancia, los vehículos disponibles en cada Tramo y los estados, suponiendo que existe una cantidad N de Tramos se lleva a cabo la siguiente expresión:

$$Vehículos_{entrantes} = \sum_{i=1}^{N} n_i \cdot I_i \cdot E_i$$

Sustituyendo los valores conocidos, la operación es la siguiente:

$$Vehículos_{entrantes} = 2 \cdot 0.5 \cdot 1 + 3 \cdot 1 \cdot 0 + 2 \cdot 0.5 \cdot 0 = 1\ vehículo$$

Por lo tanto, con los datos establecidos, se asume que 1 vehículo procederá a entrar en el Tramo.

Resumiendo, esta operación se lleva a cabo siempre y cuando haya Tramos que afecten a nuestra área de estudio. En caso contrario se establece que el Tramo de estudio es una Entrada Principal, por lo que la cantidad de vehículos de entrada se pueden extraer de la expresión del flujo de vehículos que esté asociada.

Vehículos Salientes

Si en los Vehículos Entrantes, la Importancia era una variable esencial, para los Vehículos Salientes es necesario emplear otra variable.

Los Vehículos Salientes hacen referencia al nº de vehículos situados en el punto Final ya que ellos son los transportes dispuestos a abandonar el Tramo, por lo que para descubrir esta cantidad, es necesario llevar a cabo una esquematización de los Tramos en forma de vector.

La longitud de este vector es directamente proporcional a la distancia en metros que ocupa el Tramo desde su punto inicial hasta su punto final divido por una longitud media, de modo que se calcula del siguiente modo:

$$Longitud(-) = \frac{Distancia\ (m)}{5\ \ (m)}$$

La razón de escoger 5 metros es que es un valor que no sobrepasan los vehículos predeterminados además de tener en cuenta el creciente tamaño de vehículos que defienden algunos artículos.

Por lo que un Tramo de 2 carriles de 20 metros de Distancia entre su punto inicial y final, se convierte para el planteamiento teórico en un vector de 4 celdas ($\mathrm{Longitud} = 4$).

Si este Tramo está inicialmente ($\mathrm{k}=0$) vacío y acoge el máximo que puede recibir (*2 vehículos porque tiene 2 carriles*), el vector para este Tramo se representaría del siguiente modo:

Ilustración 4: Vector vacío con dos vehículos entrantes.

Por lo tanto, si el Tramo se mantiene en Verde (*Estado activo*) recibiendo de 0 a 2 vehículos en cada ciclo, en el cuarto periodo ($k=3$) el vector se observaría de la siguiente forma:

Ilustración 5: Vector en el cuarto ciclo.

Siendo los 2 vehículos de la última celda, los Entrantes en el primer ciclo y los Salientes en el cuarto. Por ello, el tratamiento del tránsito en los Tramos y, por tanto, el cálculo de los Vehículos Salientes se realiza con el simple procedimiento de variar los valores del vector de cada Tramo, de modo que los Salientes son los vehículos de la última celda.

Conocidos los Vehículos Entrantes y Salientes se puede llevar a cabo el control de la variable de interés, es decir, el control de la cantidad de Vehículos.

Aplicación

Elementos

Dado que el proyecto en sí se centra en una aplicación para ordenador, en primer lugar, se llevaron a cabo las decisiones relacionadas con el entorno de desarrollo a utilizar por tal de crear el programa.

Tal y como un cirujano escoge la herramienta más idónea a la hora de operar, un programador debe escoger el mejor entorno de desarrollo (IDE) ya que de ello dependen la facilidad a la hora de programar y las posibilidades que pueda permitir la aplicación resultante.

Dado que se va a programar con formularios o ventanas de Windows y con un lenguaje relacionado directamente con el C, se ha escogido el Entorno de Desarrollo Visual Studio, el cual es con diferencia, el IDE más utilizado a día de hoy para aplicaciones Windows.

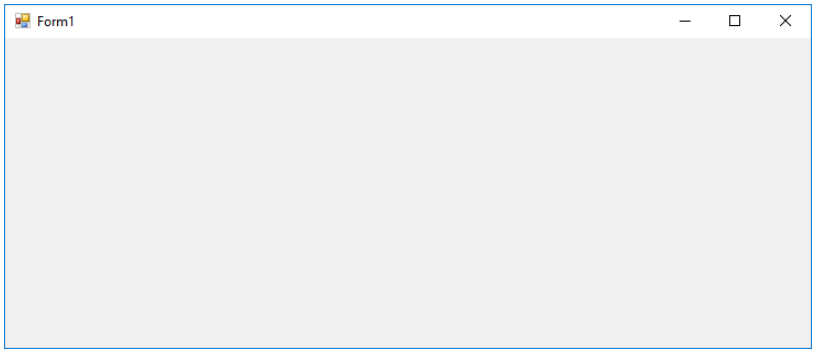
Las razones para su elección son las siguientes:

- Es posible programar mediante el lenguaje C#.
- Existen foros y documentación a consultar en caso de necesitar ayuda.
- Tiene herramientas que permiten una forma fluida de programación.
- Ofrece paquetes de extensión creados por terceros que permiten añadir nuevas características al proyecto.

Este IDE aporta además una gran cantidad de elementos que enriquecen la Interfaz de Usuario, es decir, el apartado del programa que interactúa con el usuario. De todos los elementos que ofrece Visual Studio los utilizados en el Simulador son:

- **Formulario** (*Form*): Se trata del elemento base en un proyecto de formularios para Visual Studio ya que sobre él se van a colocar los distintos elementos del proyecto. Aunque normalmente suele haber más de un formulario, en este caso solo se va a necesitar uno, el cual sufrirá cambios en su tamaño y aspecto mientras se utilice.

Ilustración 6: Ejemplo de Formulario Vacío.

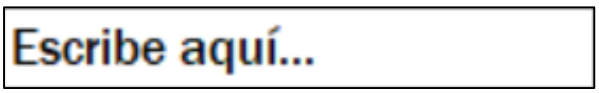


- **Botones** (*Buttons*): Elementos de control gráfico que permiten al usuario provocar un evento. Durante la simulación, todos ellos tienen aspecto de imagen tal y como se observa en esta ilustración.

Ilustración 7: Ícono del botón Cargar Simulación.

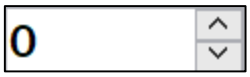
- **Cuadros de Texto** (*TextBox*): Son elementos de formularios utilizados para recibir entradas del usuario o mostrar texto. En el Simulador todos ellos se han configurado para que el texto tenga una apariencia lo más legible posible, seleccionando la fuente Franklin Gothic Medium 12 puntos. En alguno de ellos ha sido necesario programarlos para que recibirán únicamente valores numéricos.

Ilustración 8: Cuadro de Texto.



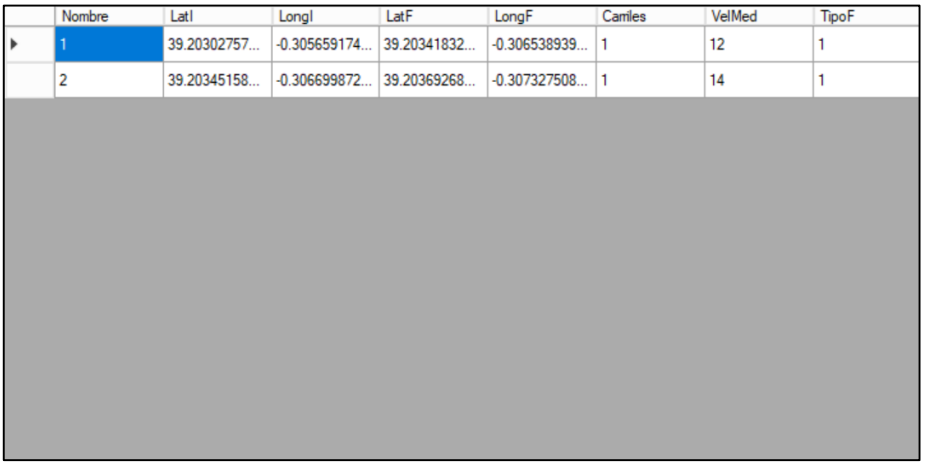
- **Seleccionador Numérico** (*NumericUpDown*): Es una combinación de cuadro de texto y un par de flechas en la que el usuario puede pulsar para seleccionar el valor deseado. Este elemento se emplea para poder definir valores como el nº de carriles, la velocidad media (Km/h) o las Importancias.

Ilustración 9: Seleccionador Numérico



- **Visualizador de Bases de Datos** (*DataGridView*): Permite al usuario visualizar el contenido de una Base de Datos. Se emplea ya que los datos más relevantes se almacenan en Tablas de Datos, de ese modo el usuario puede tener en cuenta todo aquello que se guarda.

Ilustración 10: Visualizador de Bases de Datos.



	Nombre	LatI	LongI	LatF	LongF	Carriles	VelMed	TipoF
▸	1	39.20302757...	-0.305659174...	39.20341832...	-0.306538939...	1	12	1
	2	39.20345158...	-0.306699872...	39.20369268...	-0.307327508...	1	14	1

- **Etiquetas** (*Labels*): Sirven para poder mostrar textos e imágenes no editables por el usuario, por ello, su tarea más importante es la de indicar al usuario la acción que debe realizar en cada etapa. Como todos los elementos con texto, la fuente elegida es Franklin Gothic Medium 12 puntos.

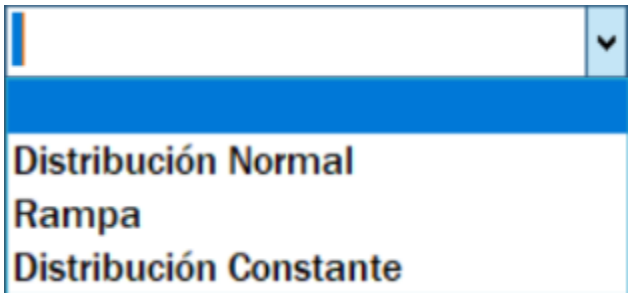
Ilustración 11: Etiqueta.

- **Cuadro de Lista** (*ListBox*): Se emplean para poder mostrar diferentes opciones en forma de lista con el objetivo de que el usuario seleccione una entre ellas. Las listas que se pueden visualizar son las referentes a Tipos de Semáforos e Importancias.

Ilustración 12: Cuadro de Lista de Tipos de Semáforos.

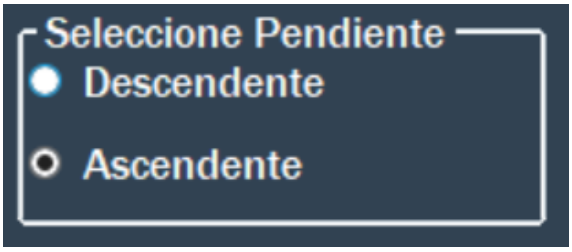
- **Cuadro Combinado** (*ComboBox*): Utilizado para poder mostrar datos en un Cuadro Desplegable de modo que se seleccione la opción que más se desee. En concreto, se utiliza para elegir el tipo de flujo de peatones y vehículos.

Ilustración 13: Cuadro Combinado de Modelo de Flujos.



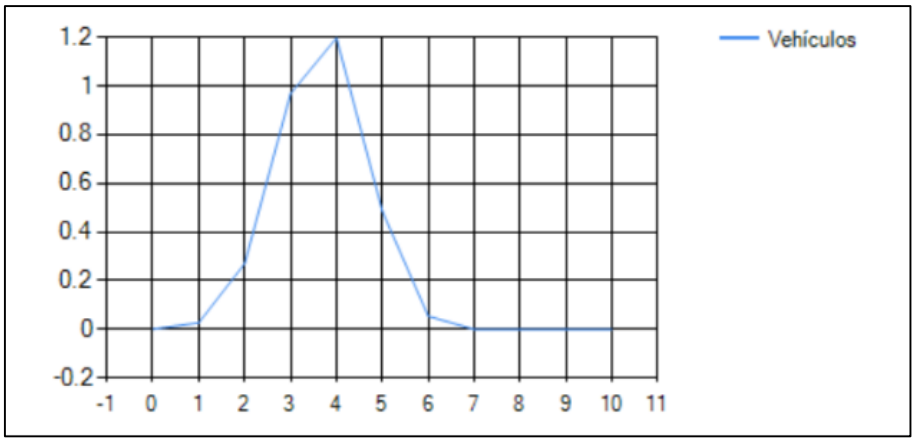
- **Botón de Opción** (*RadioButton*): Permite al usuario elegir una de las opciones conjuntamente predefinidas. En este proyecto se emplea para poder definir el signo de la expresión en Rampa.

Ilustración 14: Botones de Opción.



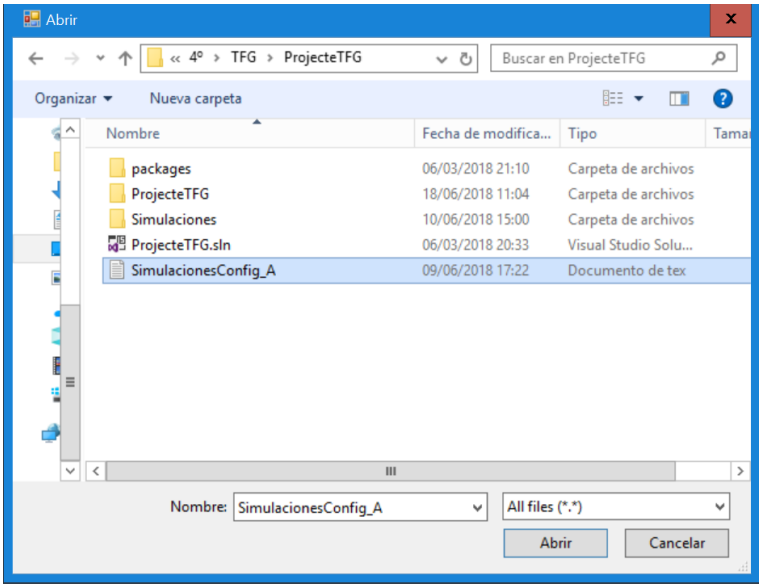
- **Gráfica** (*Chart*): Se utiliza para llevar a cabo una representación gráfica del flujo de vehículos y peatones seleccionado. La representación se lleva a cabo mediante segmentos lineales y con una variación temporal de 0 hasta once veces el Tiempo de Ciclo de cada Tramo teniendo en cuenta únicamente valores enteros.

Ilustración 15: Representación Gráfica.



Cuadro de Dialogo (*openFileDialog, folderBrowserDialog, saveFileDialog*): Empleados para poder pedir al usuario archivos a cargar (*openFileDialog*), directorios (*folderBrowserDialog*) y nombres junto con directorios para guardar archivos (*saveFileDialog*).

Ilustración 16: Cuadro Emergente.



- **Control Deslizante** (*trackBar*): Utilizado para que el usuario pueda ajustar visualmente una configuración numérica. En este caso, se emplea para variar la velocidad de simulación.

Ilustración 17: Control Deslizante.

De todos los elementos empleados en la interfaz de la aplicación, el más importante es el dedicado a la representación geográfica de los Tramos con el objetivo de observar los estados, cantidades de vehículos y peatones de cada Tramo.

Por todo ello ha sido necesario llevar a cabo una valoración de las distintas posibilidades de añadir un servicio de cartografía entre los que están:

- **Click2Map**: Aplicación de pago para crear mapas personalizados, principalmente para llevar a cabo actividades con geolocalizaciones.
- **EzMap**: Servicio online gratuito que emplea la API de Google Maps y que permite crear mapas propios añadiendo marcadores y aplicarlos en una página web al generar un código HTML.
- **GMapGIS**: Herramienta online gratuita que permite realizar cambios sobre un mapa (dibujar, superponer marcadores, medir, buscar, etc…) y generar una dirección URL con la que es posible observar este mapa personalizado.
- **GMap.NET**: Paquete de extensión de Visual Studio.

	Implementación (50 %)	Herramientas (20%)	Dificultad (15%)	Licencia (15%)	Valoración (100%)
Click2Map	8	7	10	5	7.65
EzMap	8	10	8	10	8.7
GMapGIS	6	4	7	10	6.35
GMap.NET	10	7	8	10	9.1

Tabla 2: Elementos de cartografía.

La opción de utilizar **GMapGIS** es la menos factible debido a que se trata de un programa orientado a crear mapas propios, pero de forma muy elaborada y que, además, sería muy complicado poder integrarlo en el proyecto.

Click2Map es una aplicación más sencilla de utilizar pero que sirve como una herramienta para implementar mapas en páginas Web además de tener una licencia de pago. Del mismo modo, **EzMap** es una aplicación en línea para crear mapas propios y

enlazarlos en sitios de internet, siendo en este caso gratuita pero igualmente enfocada para la Web.

Excepto **GMapGIS**, todas las aplicaciones son sencillas de utilizar, pero complicadas de implementar en un proyecto de Visual Studio, por lo que se decidió encontrar un paquete de fuente abierta para este IDE. El paquete **GMap.NET**, de libre uso es el más idóneo dado que dispone de muchos proveedores de cartografía distintos (GoogleMaps, Yahoo, ArcGIS…) además de herramientas muy útiles para poder colocar marcadores y polígonos.

- **Realización**

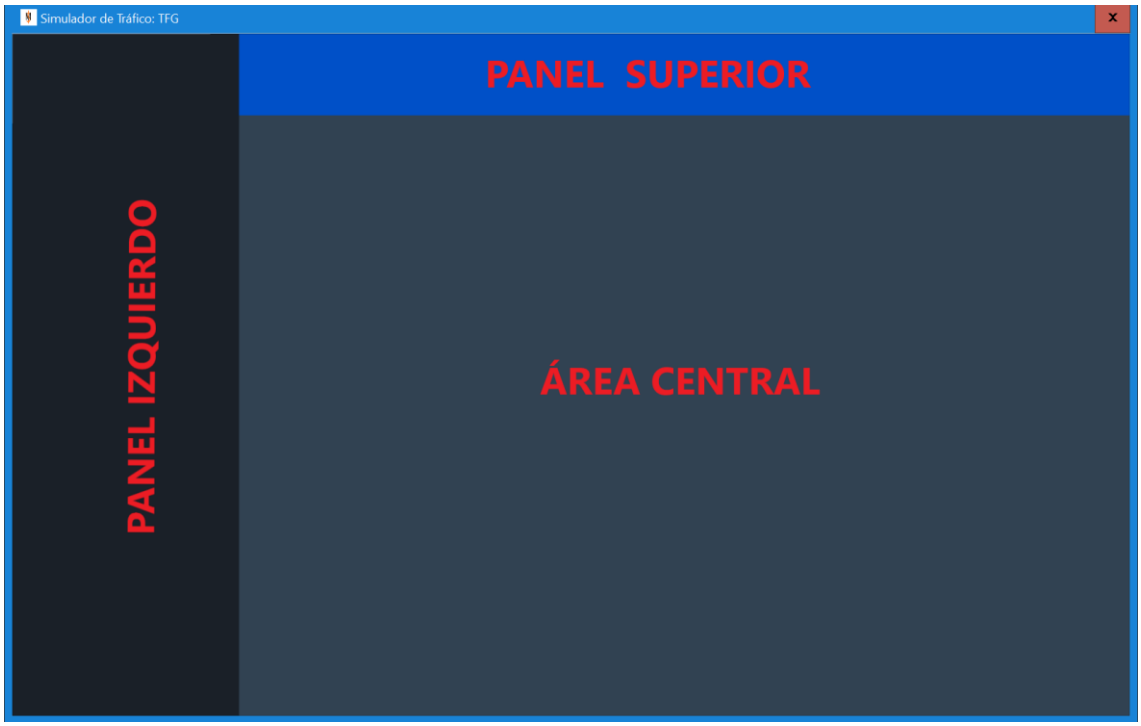
Elegidos el Entorno de Desarrollo, la fuente de mapas y demás elementos, se inicia el desarrollo del proyecto creando el formulario o ventana (*Form1*) que servirá para ubicar encima todos los elementos del simulador (etiquetas, botones, gráficas, bases de datos, mapas…).

La ventana se ha estructurado de forma que en la parte superior y en el lateral izquierdo se han ubicado paneles para poder diferenciar tres zonas, de modo que en el panel superior se dispondrán los botones de Nueva Simulación y Cargar Simulación.

En el panel izquierdo se dispondrán los botones de Continuar (necesario para poder proseguir durante todas las Etapas), Agregar/Calcular y los distintos elementos de introducción de datos. En el área central se dispondrán los elementos destinados a la representación.

Esta estructura se observa en la siguiente imagen:

Ilustración 18: Distribución de los Paneles.



De modo que, en la clase perteneciente a esta ventana, se establecen las pautas para poner y quitar los distintos elementos en las posiciones que se requiera para que el usuario pueda utilizar la aplicación de la forma más cómoda posible.

El primer paso en la utilización de este fichero es el de crear las principales variables que se utilizan para cargar, guardar o editar valores de importancia.

Por ello, se crean cuatro variables del tipo **DataTable,** es decir, Tabla de Datos. En la primera de estas variables se almacenan los valores de configuración de todos los Tramos, concretamente: Coordenadas de Inicio y Fin, Número de Carriles, el Tipo de Semáforos del Tramo, la Distancia que ocupa cada Tramo, los valores previstos de Vehículos en Entradas Principales y los valores previstos de Peatones.

La segunda variable del tipo Tabla de Datos contiene la relación entre Tramos o, mejor dicho, las Importancias de cada tramo y su suma la cual, recordando el planteamiento teórico, debe valer 0 para la consideración o no de Entradas Principales.

El tercer parámetro cambiante guarda los valores que se obtienen en cada instante de la simulación. Para un periodo x, esta variable almacena la cantidad de Vehículos que circulan en cada Tramo, además de los Peatones. También guarda los Estados de cada uno (si están Activos o Parados), la cantidad de Ciclos en el estado actual y los restantes….

La última Tabla de Datos se encarga de almacenar los valores de Vehículos y Peatones por Tramos en cada ciclo para que una vez finalizada la simulación se exporten estos valores con el objetivo de poder analizarlos.

Las Tablas de Datos son los elementos más importantes de todo el proyecto debido a que el simulador basa su funcionamiento en guardar, cargar, asignar, exportar, importar los valores que ellas contienen. Por lo que estas cuatro Tablas de Datos se utilizan a lo largo de las 10 etapas del simulador explicadas a continuación:

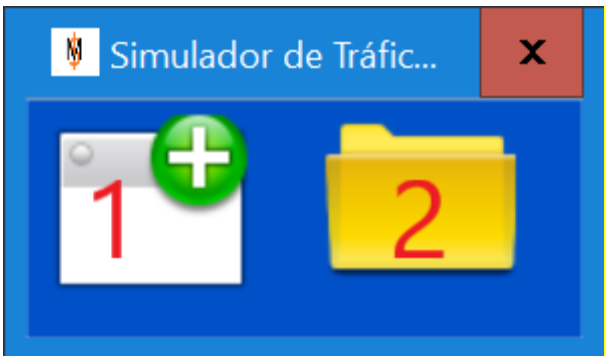
- **Etapa 0**: Pantalla Inicial.

Se trata de la primera pantalla que observa el usuario donde tendrá que escoger entre pulsar uno de los botones que realizan las siguientes opciones:

1. Inicia una nueva simulación.
2. Carga una configuración de una simulación para poder reanudarla.

La ventana se ha reducido, minimizando el panel izquierdo para poder mostrar al usuario únicamente los 2 botones.

Ilustración 19: Etapa 0.



- Etapa 1: Nombrar Proyecto.

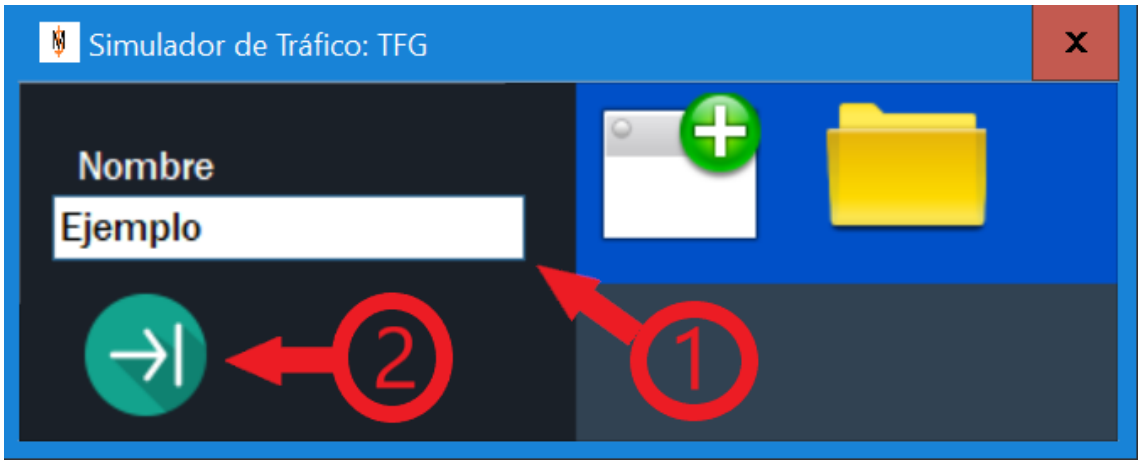
La primera fase al crear proyecto nuevo, solicita al usuario poner un nombre a la nueva configuración que se va a crear. Para ello se emplea:

1. Un Cuadro de Texto (*TextBox*) donde escribir el nuevo Nombre de Proyecto.
2. El Botón de Continuar (*Button*).

Para llevar a cabo esta fase, se amplía el panel izquierdo y se aumenta el tamaño de la ventana constantemente cada 50 ms mediante un temporizador con el objetivo de llevar a cabo una animación, de modo que la ventana alcanza un tamaño de 450x180 píxeles.

Se muestran los elementos 1 y 2, además de una Etiqueta con el texto: “Proyecto:”, de modo que se hace intuir al usuario que debe escribir el Nombre del Proyecto.

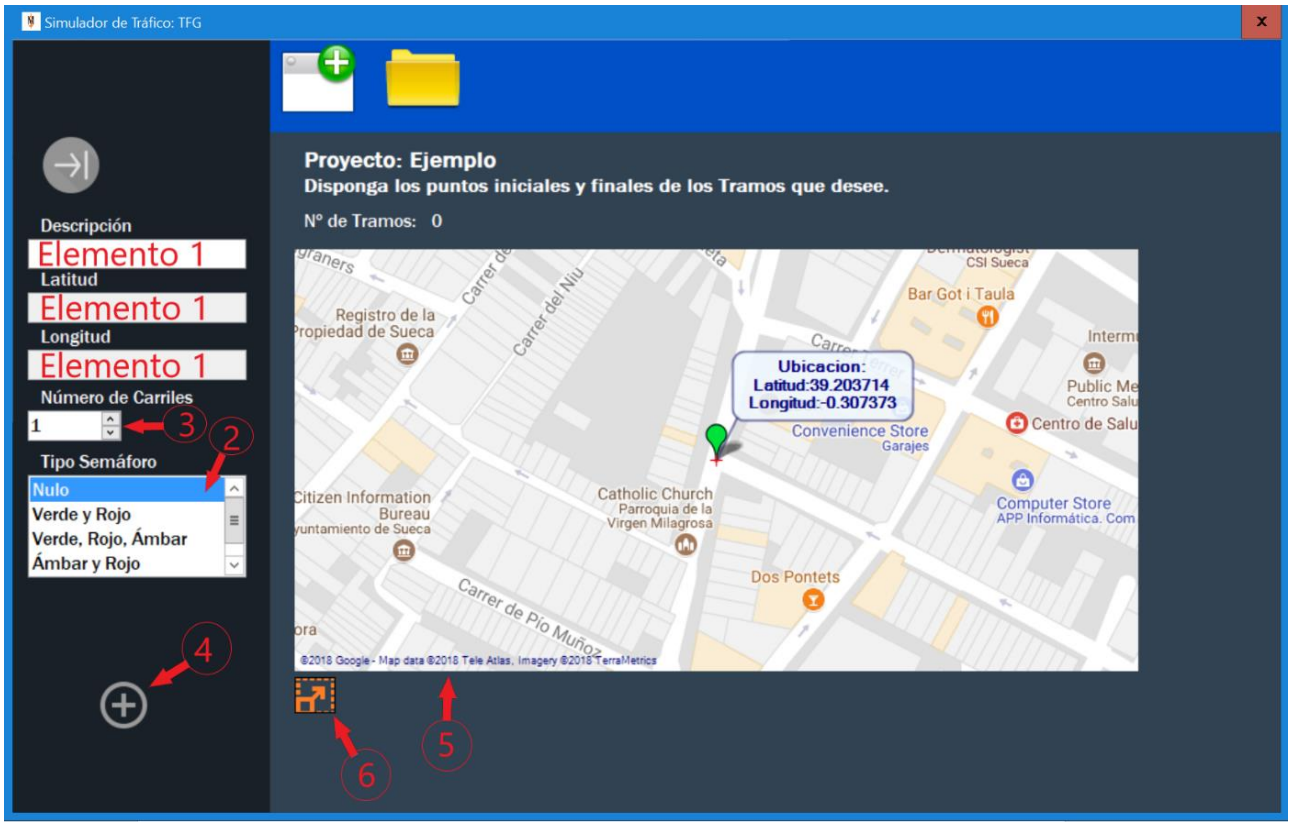
Ilustración 20: Etapa1.



- **Etapa 2**: Disposición de los Tramos.

En esta parte del programa el usuario señala el punto Inicial y Final de los distintos Tramos, la cantidad de Carriles y el Tipo de Semáforo en cada uno de ellos.

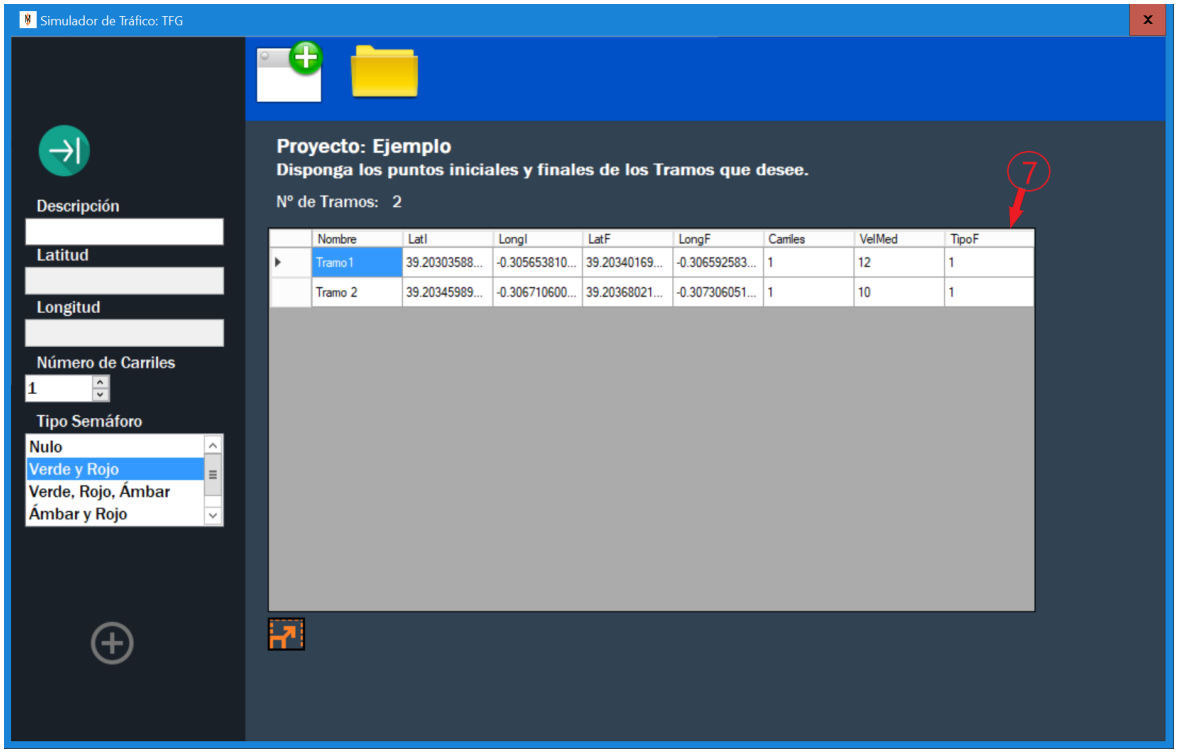
Ilustración 21: Etapa 2 (Elementos).



El desarrollo de esta etapa se ha basado en ubicar los elementos de Interfaz en posiciones adecuadas. Estos elementos son:

1. Tres Cuadros de Texto (*TextBox*).
2. Cuadro de Lista (*ListBox*).
3. Un seleccionador numérico (*NumericUpDown*).
4. Botón Agregar (*Button*).
5. Mapa (*gMapControl*).
6. Botón para cambiar de vista (*Button*).
7. Visualizador de Bases de Datos (*dataGridView*).

Ilustración 22: Etapa 2 (Visualización de los datos).



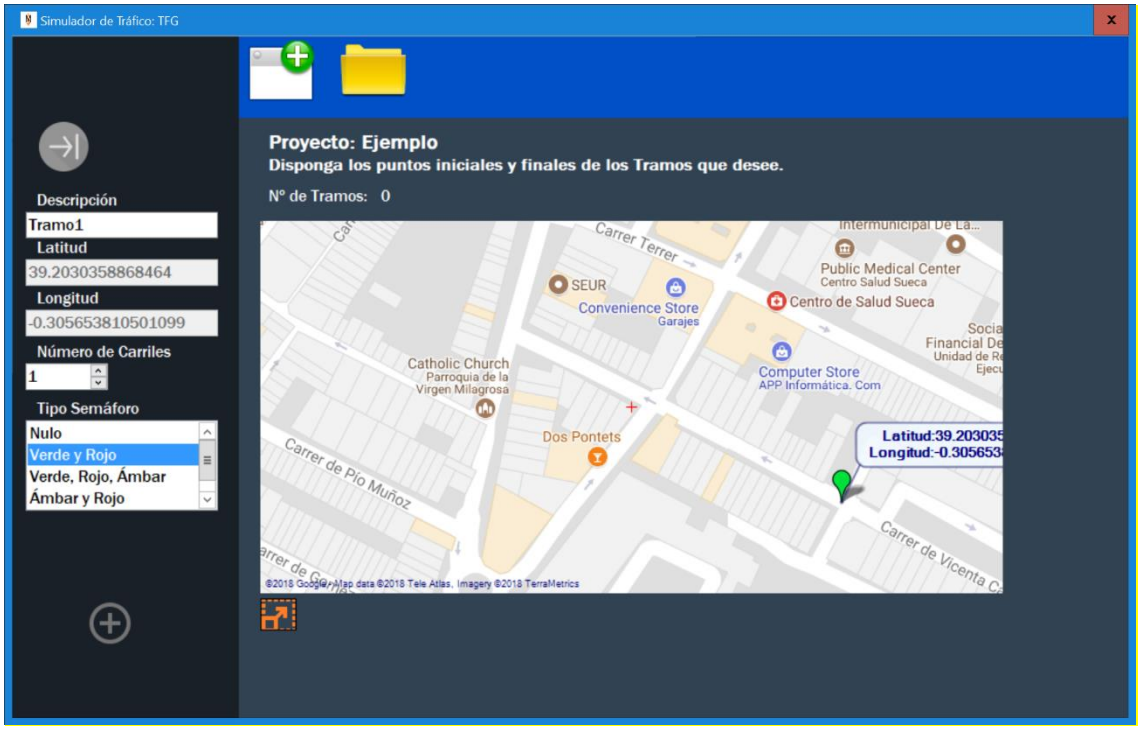
En el lateral izquierdo desaparece el Cuadro de Texto y la Etiqueta de la Etapa 1 para que aparezcan por pantalla cinco Etiquetas informando de los datos que deben ser dispuestos en los Cuadros de Texto (*Coordenadas y Nombre de Tramo*), en el Cuadro de Lista (*Tipo de Semáforo en el Punto Final*) y el Seleccionador Numérico (*nº de Carriles*).

En el centro se añaden 3 etiquetas donde se muestra el Nombre del Proyecto, la acción a realizar y la cantidad de Tramos dispuestos por el usuario.

En último lugar se agrega el Mapa solapado al Visualizador de Bases de Datos con un Botón para poder alternar entre ambos (*Elemento 7*) y de este modo poder ahorrar espacio.

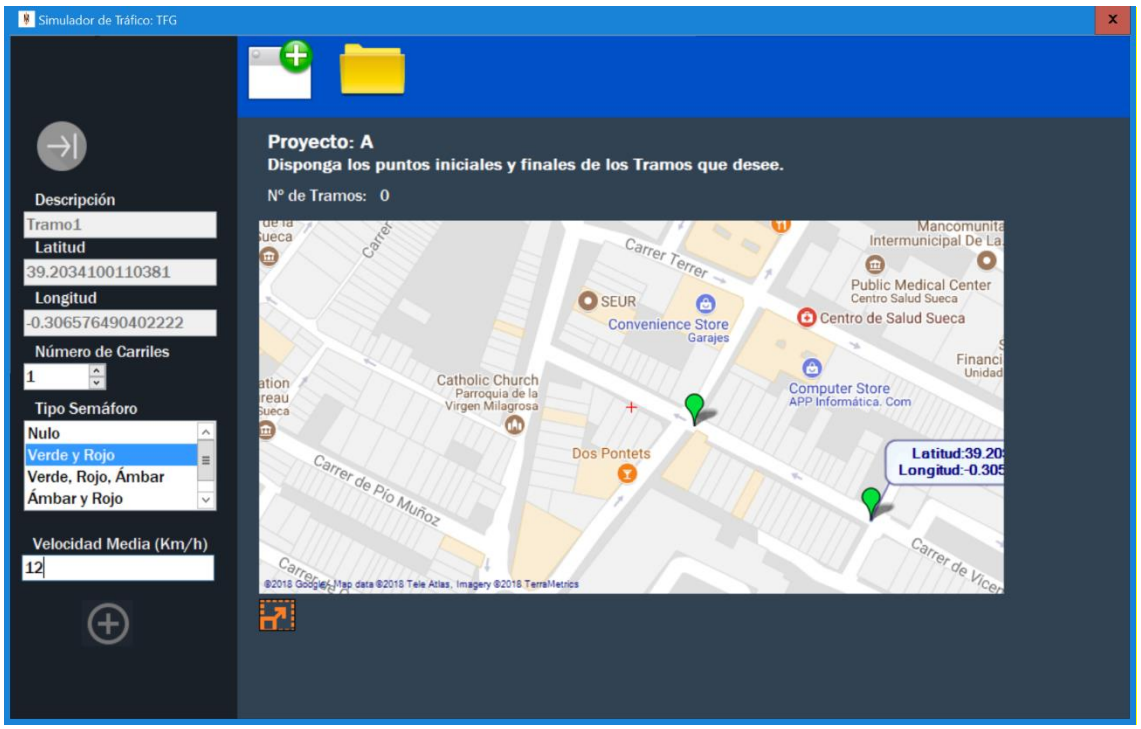
Pulsando dos veces sobre el Mapa se rellenan automáticamente los Cuadros de Texto de las Coordenadas, por lo que en esta etapa el usuario únicamente ha de pulsar dos veces con lo que se crea un marcador y se llenan los Cuadros de Texto con las Coordenadas. Además, se debe escribir el Nombre del Tramo, seleccionar el Tipo de Semáforo y el nº de Carriles con lo cual el aspecto será el siguiente:

Ilustración 23: Etapa 2 (Punto Inicial).



Una vez se pulse el Botón Agregar/Calcular (elemento 4) se muestra un nuevo Cuadro de Texto junto con una etiqueta con el texto “Velocidad Media (Km/h)”.

Ilustración 24: Etapa 2 (Punto Final).



Para poder continuar el usuario debe introducir la Velocidad, pulsar dos veces seguidas sobre el punto del Mapa y pulsar el Botón Agregar/Calcular.

Este es el procedimiento para agregar cada uno de los Tramos de la simulación. Cada vez que se pulsa el Botón Agregar/Calcular se almacena la información introducida en la Tabla de Datos 1 que guarda toda la configuración excepto los datos relacionados con las Importancias.

Una vez creados al menos dos Tramos, se puede emplear el botón Continuar de la Etapa 1 para poder avanzar a la siguiente Etapa.

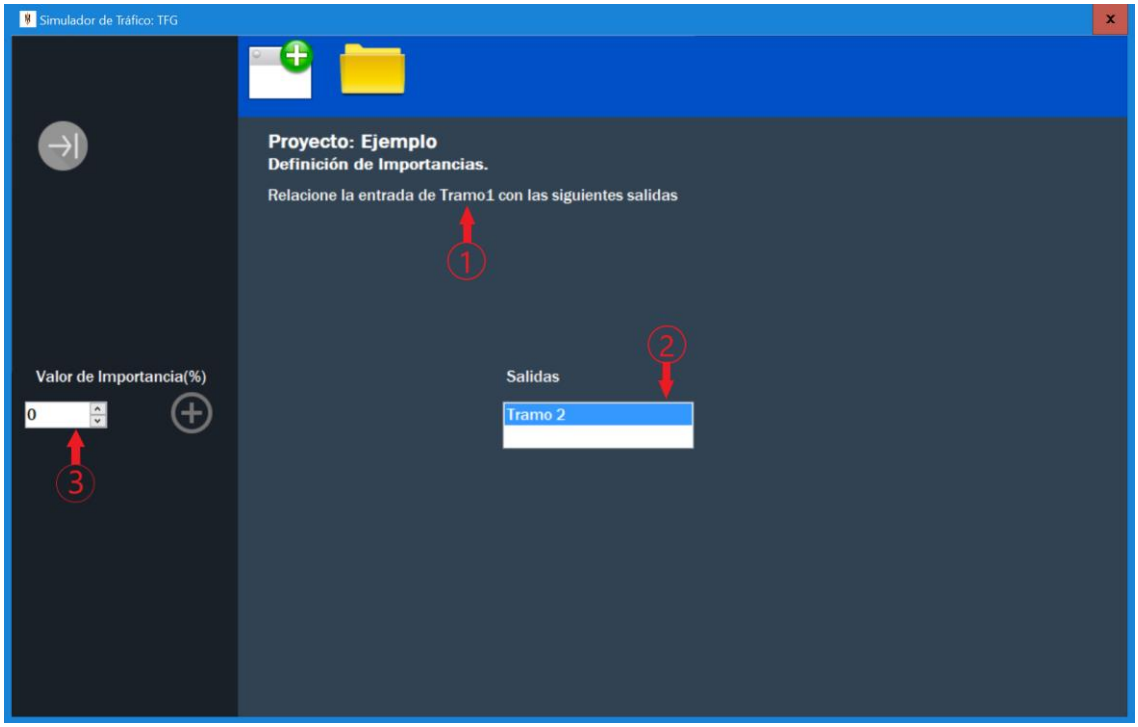
- **Etapa 3**: Relación de Importancias.

Durante esta etapa desaparecen los elementos anteriores manteniéndose las Etiquetas de la Zona Central, se modifican las instrucciones y el Seleccionador Numérico (*Elemento 3 de la Etapa 2*) junto con su Etiqueta.

Además, se ubica un Cuadro de Lista donde aparecen todos los Tramos (excepto el que se está preguntando en las instrucciones) y la Etiqueta adjunta con el texto “Salidas”.

Por lo tanto, durante esta Etapa el usuario estipula los valores de Importancias para el Tramo que la aplicación le indica en la Etiqueta (*Elemento 1*), en cada una de las salidas del Cuadro de Lista (*Elemento 2*). El valor de Importancia que predeterminadamente es 0, se puede variar en el Seleccionador Numérico (*Elemento 3*).

Ilustración 25: Etapa 3.



- **Etapa 4**: Flujo de Vehículos en Entradas Principales.

Para los Tramos indicados por la aplicación, que son únicamente los considerados como Entradas Principales, se pide una previsión de flujo de vehículos, es decir, cantidad de vehículos en función del tiempo, o como es en este caso, por ciclos.

La aplicación únicamente pedirá esta información para los Tramos considerados como Entradas Principales debido a que en la etapa anterior se calculó la suma de las Importancias indicadas por el usuario, de forma que aquellas sumas cuyo valor son 0, señalan que sus Tramos son Entradas Principales (tal y como se explica en el planteamiento teórico).

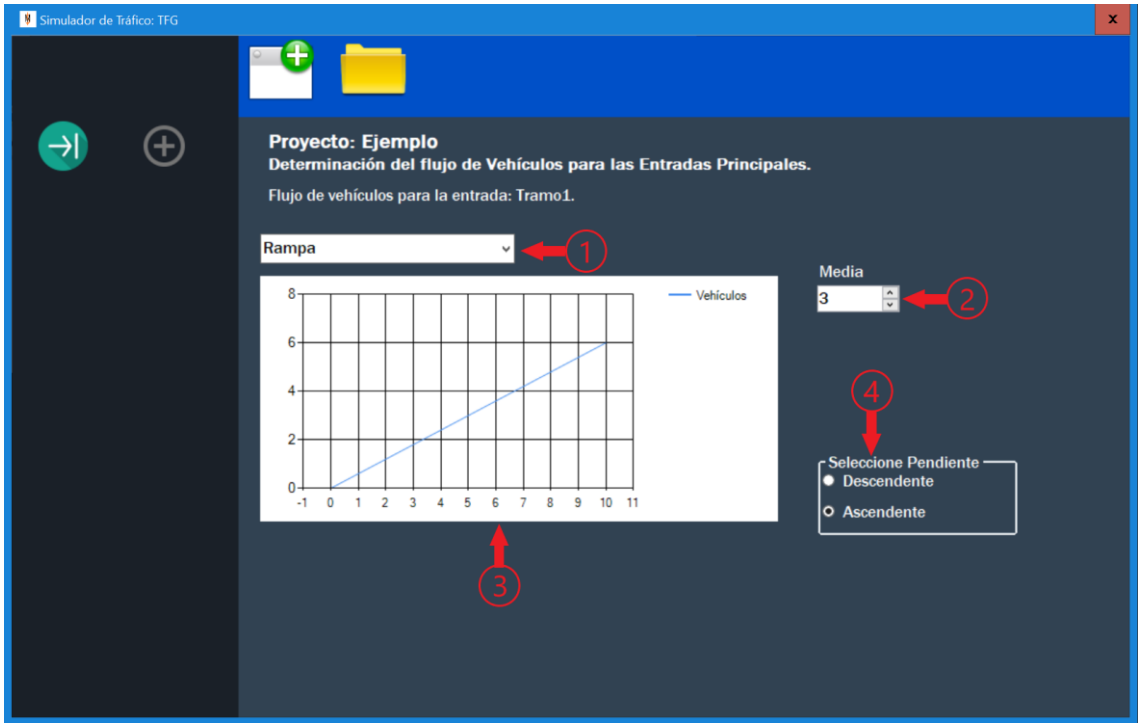
Para obtener el modelo a seguir que representa el flujo, se permite la elección del modelo a los usuarios entre Distribución Normal, Constante o en Rampa, además de los valores característicos. De forma que el resultado del modelo para el tiempo correspondiente a 10 ciclos de duración de cada Tramo se puede observar en la gráfica.

Desaparecen los elementos previos excepto las Etiquetas de instrucciones y del Nombre de Proyecto. Posteriormente se muestran los siguientes ítems:

1. Seleccionador desplegable de los distintos modelos a seguir.
2. Seleccionador/es numérico/s para determinar los valores significantes del modelo seleccionado.
3. Grafica que muestra el resultado del modelo seleccionado para 10 ciclos.
4. Botón de Opción para determinar el signo en caso de seleccionar la expresión en Rampa.

Una vez seleccionados los valores y el tipo de expresión, al pulsar el botón Agregar/Calcular se lleva a cabo la representación gráfica que contiene los valores del flujo de vehículos del cual, durante la simulación, únicamente se tendrán en cuenta valores enteros.

Ilustración 26: Etapa 4.



Se pueden hacer tantas representaciones gráficas como se quieran dado que se muestra y se guarda (en la Tabla de Datos 2) la configuración al pulsar el Botón Agregar/Calcular. Pulsar el Botón Continuar sirve para proseguir a la siguiente etapa o a la selección de flujo de vehículos de la siguiente Entrada Principal.

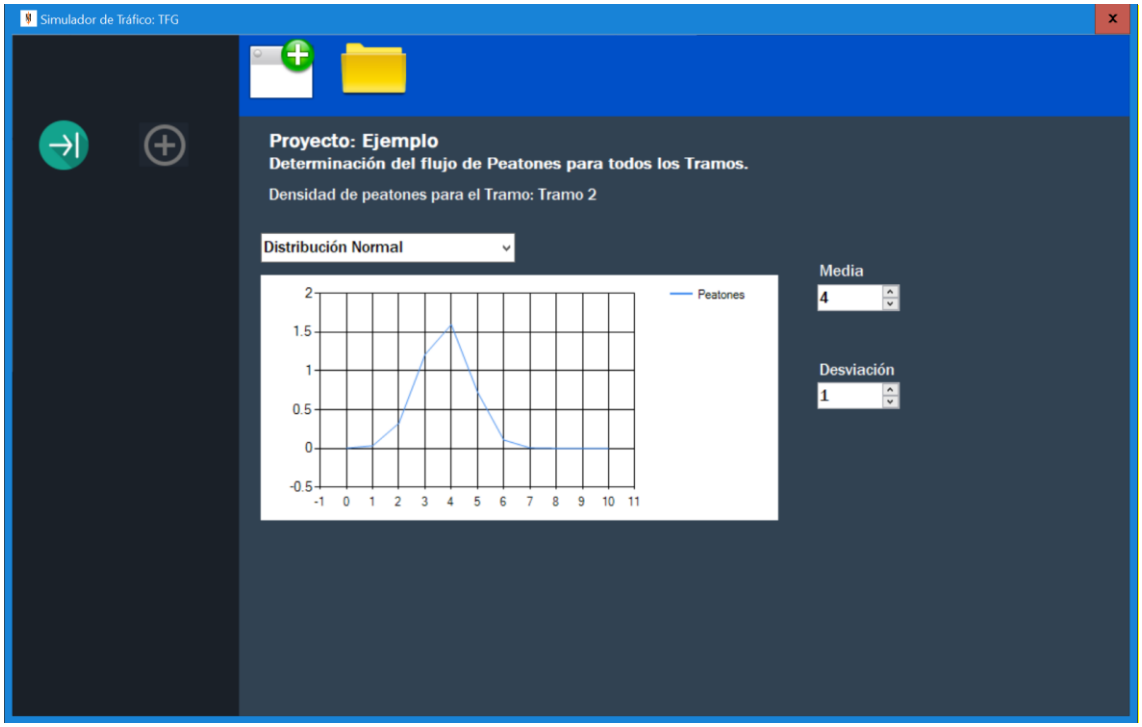
- Etapa 5: Flujo de Peatones.

La aplicación solicita al usuario que indique los valores previstos de peatones para todos los Tramos en función del tiempo por lo que el procedimiento es el mismo que en la etapa anterior pero más extensa al tratarse de definir valores para todos los Tramos.

La indicación se lleva a cabo del mismo modo que con los vehículos, es decir, señalando la expresión que más se adecua (Distribución normal, Constante o en Rampa) y los valores de esta expresión. El resultado se puede observar en una gráfica que muestra los 10 ciclos de duración del Tramo.

Los elementos de esta ventana son los mismos de la Etapa anterior.

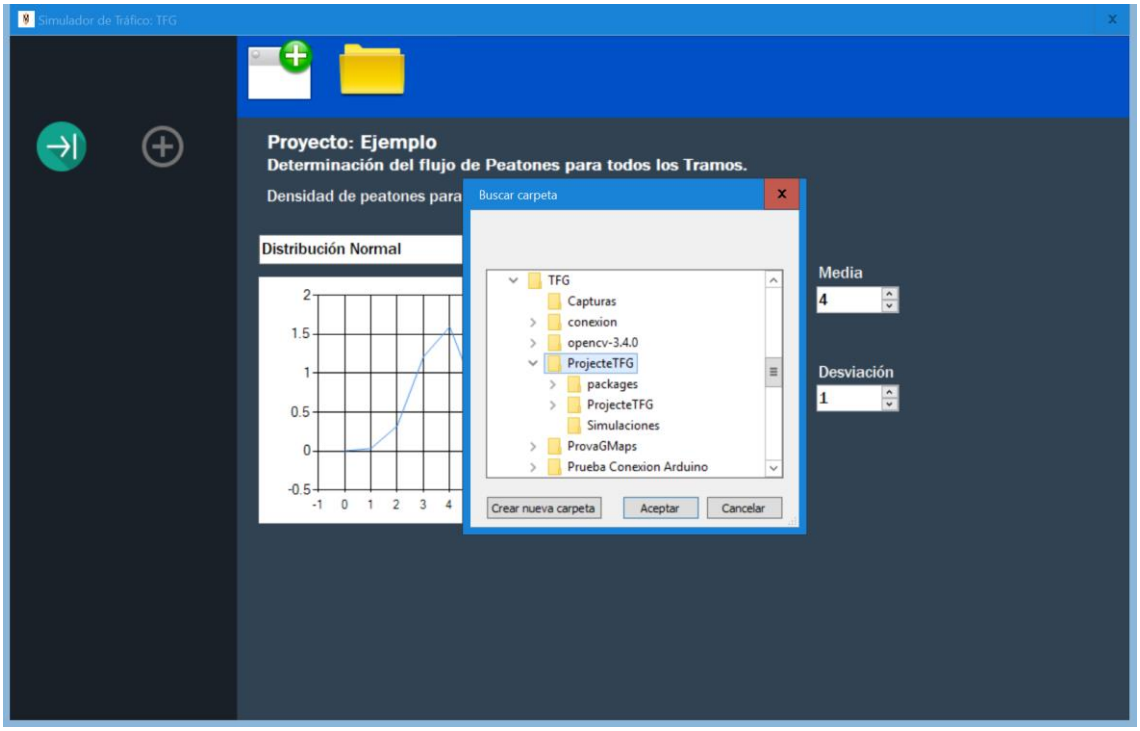
Ilustración 27:Etapa 5.



- Etapa 6: Guardado de la Configuración.

Nada más terminar la etapa anterior el programa pide al usuario que seleccione una carpeta donde guardar la configuración. Es decir, todos los valores que se han estado detallando desde la Etapa 2 a la Etapa 5 y que hacen referencia a las dos primeras variables Tablas de Datos (Tramos e Importancias).

Ilustración 28: Etapa 6.



Por lo que se emplea el Control folderBrowserDialog que provoca el Cuadro Emergente que al recibir la aprobación del usuario, la aplicación crea un archivo .txt que se encarga de almacenar todos los detalles del proyecto en el archivo, nombrándolo "Config_(Nombre del Proyecto)". Donde (Nombre del Proyecto) es el texto escrito en la Etapa 1.

Ilustración 29: Contenido archivo de configuración.

```
Ejemplo
NOMBRES
Tramo1
Tramo2
Tramo3
Tramo4
LATITUDES INICIALES
39.468222,39.468472,39.467889,39.467889
LONGITUDES INICIALES
-0.373889,-0.372944,-0.373722,-0.37375
LATITUDES FINALES
39.467917,39.467917,39.467806,39.467833
LONGITUDES FINALES
-0.37375,-0.373722,-0.373694,-0.373833
VELOCIDADES KM/H
15,22,8,12
TIPOS
1,2,3,2
CARRILES
2,3,1,3
DISTANCIAS
0.0359924299891593,0.0910334564203908,0.00954770143576387,0.00947297761366822
DENSIDAD VEHICULOS
1,3,1
```

```
1,5,1


DENSIDAD PEATONES
2,1,0
1,4,1
2,1,0
2,4,1
IMPORTANCIAS
0,0,0,0
0,0,0,0
50,30,0,0
50,70,0,0
SUMA IMPORTANCIAS
0,0,80,50
```

- **Etapa 7**: Simulación.

En esta fase de la aplicación, aparecen en pantalla el Mapa con los Tramos especificados durante la Configuración, esta vez con sus puntos Iniciales y Finales unidos por una flecha y un marcador en medio.

El color de los marcadores inicialmente es blanco dado que la simulación todavía no ha empezado, aunque una vez iniciada, el color cambiará en función del estado de cada Tramo, representando de esta forma, el color que tendría cada semáforo.

Durante la simulación, es posible observar valores importantes de cada Tramo con mantener el ratón encima de cualquier marcador. De esta forma se puede llevar a cabo un seguimiento por parte del usuario.

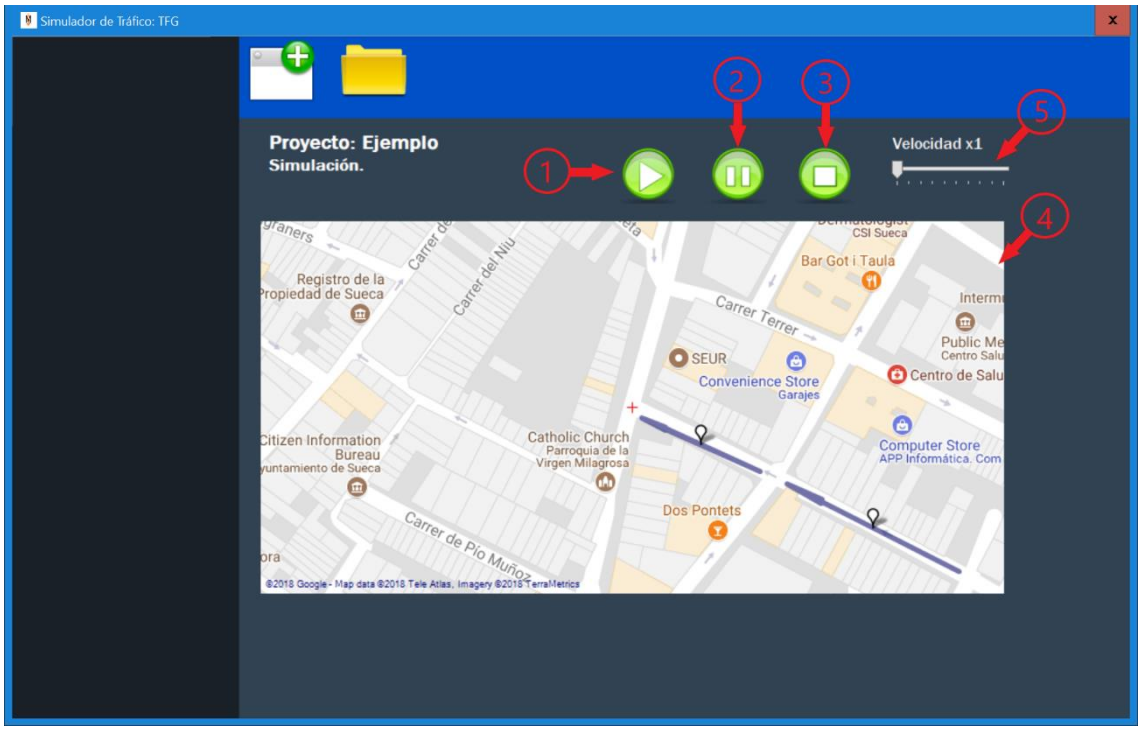
La flecha y su dirección reflejan el sentido de la circulación del Tramo por lo que únicamente es una herramienta visual para ayudar al usuario a entender más fácilmente la simulación.

En resumen, para poder controlar la simulación el usuario dispone de los siguientes elementos:

1. Botón Iniciar, para empezar a simular una vez se hayan cargado todos los Tramos sobre el mapa.
2. Botón Pausa, para congelar la simulación de forma que se puede volver a activar con el Botón Iniciar.
3. Botón Parar, que termina la simulación.
4. Mapa interactivo, que contiene toda la información ya explicada.

5. Control Deslizante (*trackBar*) con el que se regula la velocidad de Simulación.

Ilustración 30: Etapa 7 (Elementos).



Para llevar a cabo todo el procedimiento de esta Etapa se ha creado una clase, es decir, un fichero titulado Algoritmo con las funciones necesarias para llevar a cabo el cambio de estados, el cálculo de Peatones, Vehículos Entrantes y Salientes de cada Tramo resolviendo por tanto la cantidad de Vehículos y el desplazamiento de los vectores que representan cada Tramo.

De todas las funciones creadas en el fichero Algoritmo, la más importante es la función Operaciones, la cual recibe las variables definidas por el usuario y la tercera Tabla de Datos, donde se guarda toda la información a representar en el Mapa.

Por lo tanto, la función Operaciones recibe la tercera Tabla de Datos, lleva a cabo los cambios de Estado, los cálculos de Vehículos, Peatones y los desplazamientos de vectores con la ayuda de otras funciones definidas en el mismo fichero, para actualizar al final la tercera Tabla de Datos con los nuevos valores de todos los Tramos a representar.

Tal y como se ha explicado anteriormente, el proceso se realiza mediante periodos de un reloj. Por lo que cada vez que transcurre un ciclo, se activa la función Operaciones y se actualiza la Tabla de Datos y por lo tanto el Mapa.

Cada vez que transcurre un ciclo, los Tramos avanzan un tiempo real equivalente al tiempo que tarda un coche en recorrer 5 metros con la velocidad media más alta de todos los Tramos, debido a que al tener en cuenta la velocidad más alta, el tiempo va a ser el menor y por tanto todos los Tramos funcionaran a la par.

Es decir, si hay dos Tramos (A y B) con 12 Km/h y 10 Km/h de velocidades media respectivamente, la operación que se lleva a cabo es la siguiente:

$$T_A = \left(12\frac{\cancel{Km}}{\cancel{h}} \cdot \frac{1000\cancel{m}}{1\,\cancel{Km}} \cdot \frac{1}{5\cancel{m}} \cdot \frac{1\cancel{h}}{3600s} \right)^{-1} = 1.5\ s$$

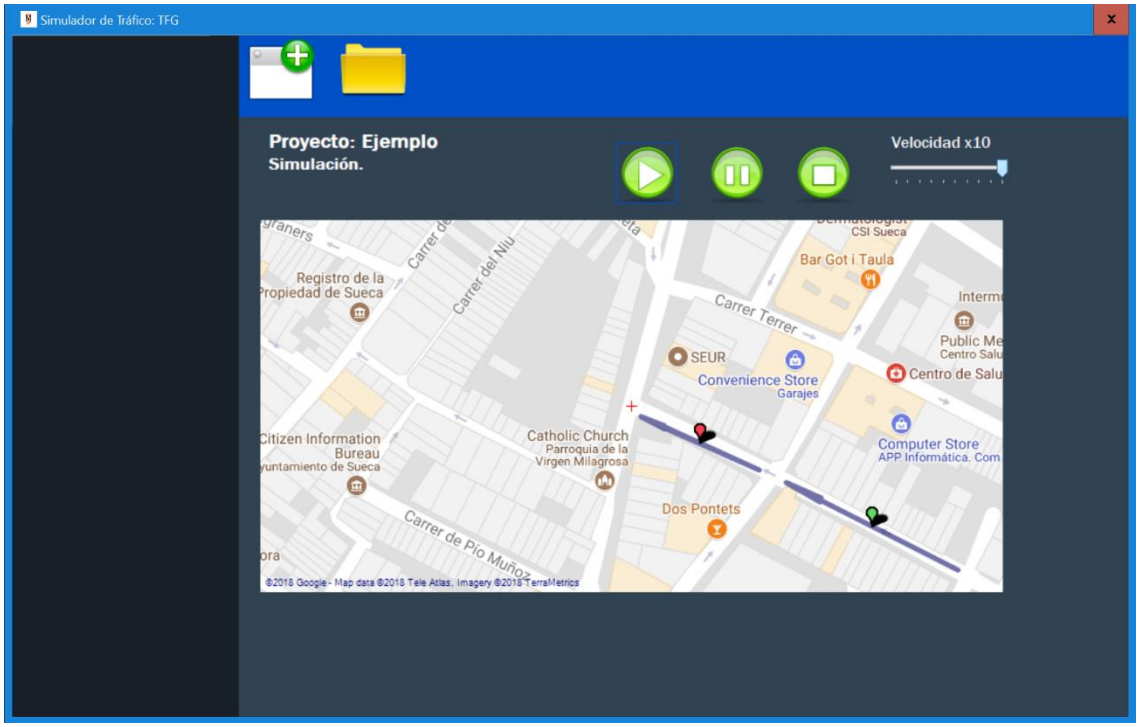
$$T_B = \left(10\frac{\cancel{Km}}{\cancel{h}} \cdot \frac{1000\cancel{m}}{1\,\cancel{Km}} \cdot \frac{1}{5\cancel{m}} \cdot \frac{1\cancel{h}}{3600s} \right)^{-1} = 1.8\ s$$

Sabiendo que el Tramo A ocupa el menor tiempo (debido a que tiene la mayor velocidad media), en cada ciclo de reloj la simulación avanza un tiempo virtual de 1.5 segundos.

Como anteriormente, informar de que para realizar esta operación se ha tenido en cuenta una distancia de 5 metros debido a la longitud media creciente de los vehículos.

En último lugar, una vez actualizada la tercera Variable, los nuevos datos de interés (Vehículos, Peatones …) de cada Tramo en cada ciclo, se guardan en la cuarta Tabla de Datos, junto con otros valores constantes de interés, como puede ser la Capacidad, el Tiempo Restante o la Iteración (ciclo de reloj transcurrido).

Ilustración 31: Etapa 7 (Simulación).



- **Etapa 8**: Exportación.

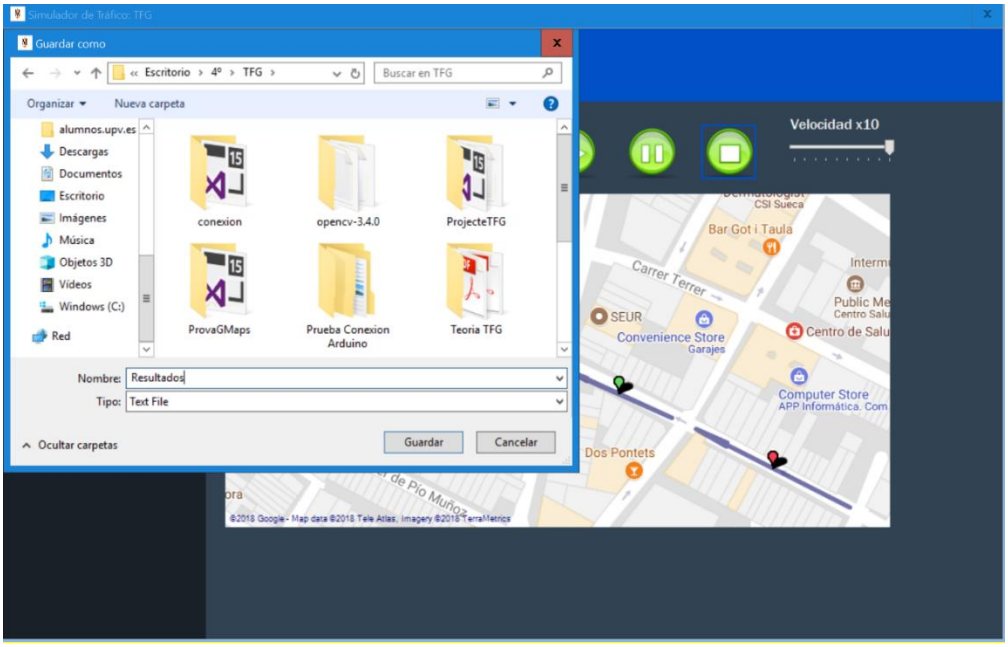
Una vez el usuario finalice la simulación, aparece en pantalla un Cuadro Emergente que solicita un nombre y una carpeta, de forma que, con esta información, la aplicación crea un fichero .txt con el nombre y en la carpeta indicada.

En este fichero .txt quedan almacenados los valores obtenidos de Vehículos, Peatones y Estados de todos los Tramos en cada uno de los ciclos que ha durado la simulación.

Por lo tanto, los valores que se exportan son los que ocupan la cuarta variable Tabla de Datos que almacena los resultados de la simulación.

Para llevar a cabo este proceso, se ha necesitado emplear el Control saveFileDialog, creando un archivo de Texto y escribiendo línea por línea toda la información de la cuarta Tabla de Datos por Tramos

Ilustración 32: Etapa 8.

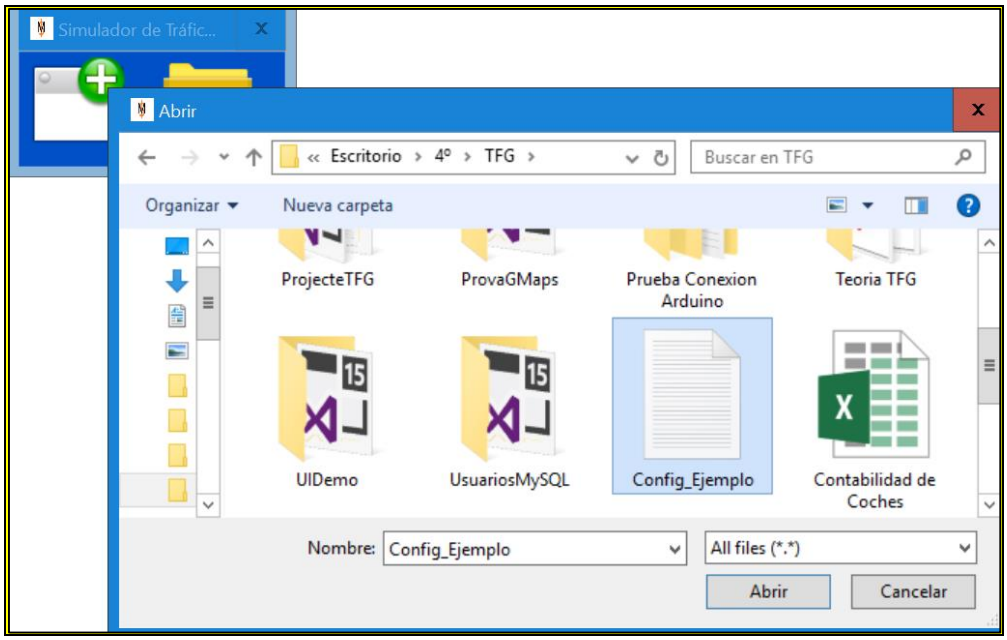


- **Etapa 9**: Importación.

La última fase disponible se encarga de obtener cualquier configuración a partir de los archivos creados en la Etapa 6, es decir, archivos de extensión .txt que contienen toda la información expuesta por el usuario desde la Etapa 1 hasta la Etapa 5.

De este modo, es posible llevar a cabo simulaciones sin necesidad de tener que repetir todo el proceso de disposición de Tramos, Importancias, etc…

Ilustración 33: Etapa 9.



Para poder llevar a cabo esta Etapa se emplea el Control openFileDialog en el Botón de Carga (*Elemento 2 Etapa 0*).

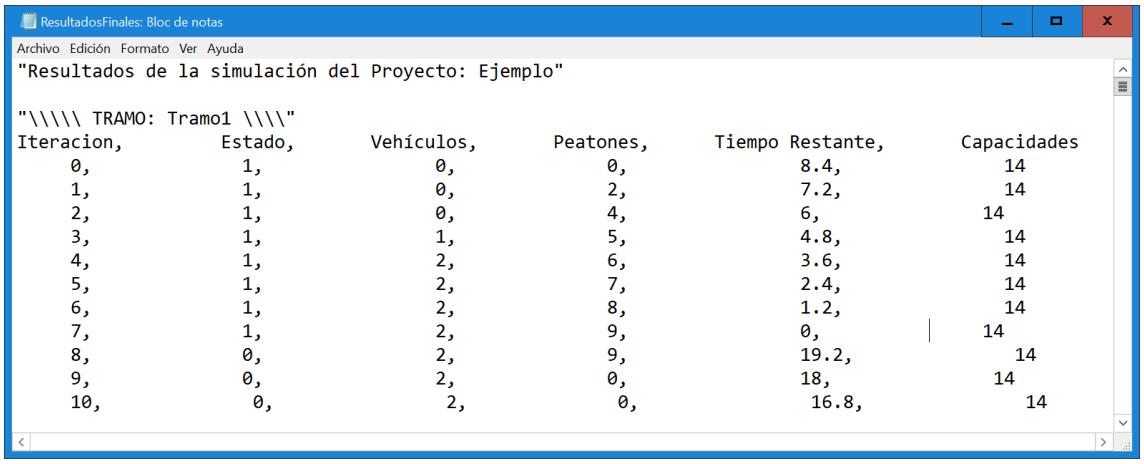
- **Etapa 10**: Salida.

En la Etapa 8 del Simulador, tal y como se ha explicado, se lleva a cabo la creación de un Archivo de Texto donde para cada Tramo se escriben los siguientes valores:

- **Iteración**: Veces que se ha llamado la función Operaciones de la clase Algoritmo.
- **Estado**: Situación de los Tramos. Si su valor es verdadero, significa que está en Verde o en Ámbar, en cambio, con valor falso, el semáforo del Tramo está en Rojo.
- **Vehículos**: Cantidad de Vehículos situados en cada Tramo en cada iteración.
- **Peatones**: Número de Peatones en cada Tramo.
- **Tiempo Restante**: Tiempo en segundos que restan para que cada Tramo cambie de Estado.
- **Capacidad**: Número de Vehículos que pueden ocupar un Tramo determinado.

Las variables detalladas se observan en los Archivos de Texto de la siguiente forma:

Ilustración 34: Etapa 10 (Contenido del archivo de resultados).



Tal y como se explicó en el funcionamiento de la Etapa 7, cada Tramo avanza una cantidad de tiempo distinta según avanzan las Iteraciones, por ello para algunos valores de Iteración, los valores de un determinado Tramo no varían.

A pesar de que pueda suponer un problema, la representación gráfica y por tanto el análisis de los resultados es perfectamente posible.

Una forma de llevar a cabo esta representación se basa en el empleo de una hoja de Excel, donde se importan Datos desde el Archivo de Texto gracias a la delimitación por Comas de los valores imprimidos en los archivos.

Ilustración 35: Etapa 10 (Resultados importados a una hoja de Excel).

	A	B	C	D	E	F	G
1	Resultados de la simulación del Proyecto: Ejemplo						
2							
3	\\\\\ TRAMO: Tramo1 \\\\						
4	Iteracion	Estado	Vehículos	Peatones	Tiempo Restante	Capacidades	1.2
5	0	1	0	0	8.4	14	0
6	1	1	0	2	7.2	14	1.2
7	2	1	0	4	6	14	2.4
8	3	1	1	5	4.8	14	3.6
9	4	1	2	6	3.6	14	4.8
10	5	1	2	7	2.4	14	6
11	6	1	2	8	1.2	14	7.2
12	7	1	2	9	0	14	8.4
13	8	0	2	9	19.2	14	9.6
14	9	0	2	0	18	14	10.8
15	10	0	2	0	16.8	14	12
16	11	0	3	0	15.6	14	13.2
17	12	0	4	0	14.4	14	14.4
18	13	0	4	0	13.2	14	15.6
19	14	0	4	0	12	14	16.8
20	15	0	4	0	10.8	14	18
21	16	0	4	0	9.6	14	19.2
22	17	0	4	0	8.4	14	20.4
23	18	0	4	0	7.2	14	21.6
24	19	0	4	0	6	14	22.8
25	20	0	4	0	4.8	14	24
26	21	0	4	0	3.6	14	25.2
27	22	0	5	0	2.4	14	26.4
28	23	0	6	0	1.2	14	27.6
29	24	0	9	0	0	14	28.8
30	25	1	9	0	49.2	14	30
31	26	1	7	1	48	14	31.2
32	27	1	5	2	46.8	14	32.4
33	28	1	4	3	45.6	14	33.6
34	29	1	4	4	44.4	14	34.8
35	30	1	3	4	43.2	14	36
36	31	1	2	4	42	14	37.2
37	32	1	2	4	40.8	14	38.4
38	33	1	2	6	39.6	14	39.6
39	34	1	2	8	38.4	14	40.8
40	35	1	1	10	37.2	14	42
41	36	1	0	11	36	14	43.2
42	37	1	0	12	34.8	14	44.4
43	38	1	0	13	33.6	14	45.6
44	39	1	1	14	32.4	14	46.8

Nota: Imagen obtenida mediante una hoja de Excel 2016.

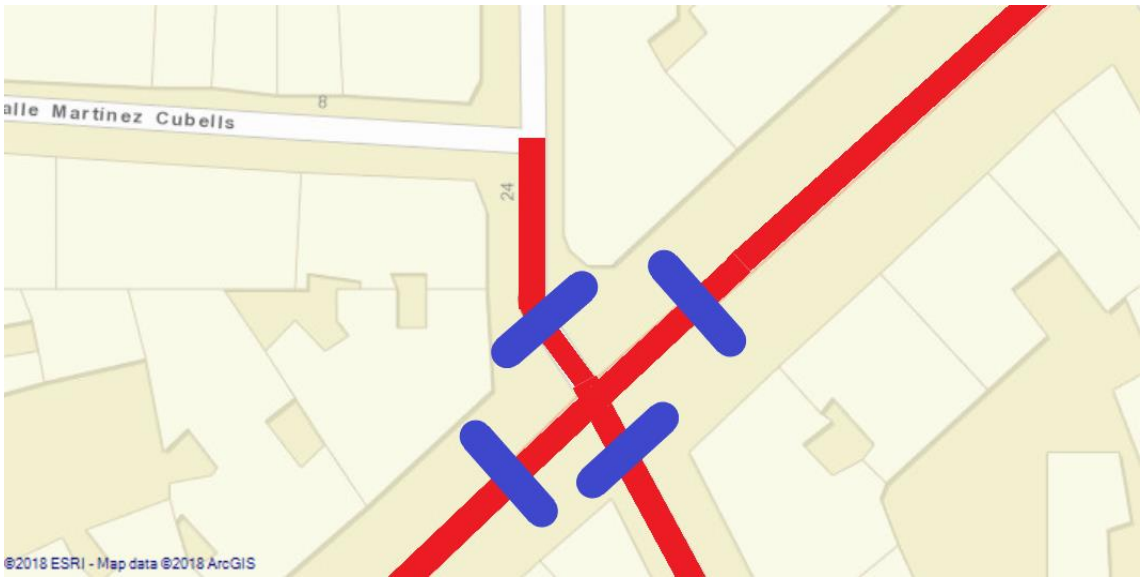
Pruebas Realizadas

Dentro la ciudad de Valencia, existe un pequeño cruce donde observar el funcionamiento del proyecto.

Se trata de la intersección entre la Calle Coló, la Calle de Pascual y Genís además de la Calle Félix Pizcueta (39°28'04.4"N 0°22'25.5"O), la cual está regulada mediante semáforos en cada una de las entradas y salidas, por lo tanto, es un área apropiada para llevar a cabo el planteamiento teórico realizando los siguientes pasos:

- Tener en cuenta las zonas afectadas (*en Rojo*) y la ubicación de los semáforos (*en Azul*).

Ilustración 36: Zonas Afectadas.



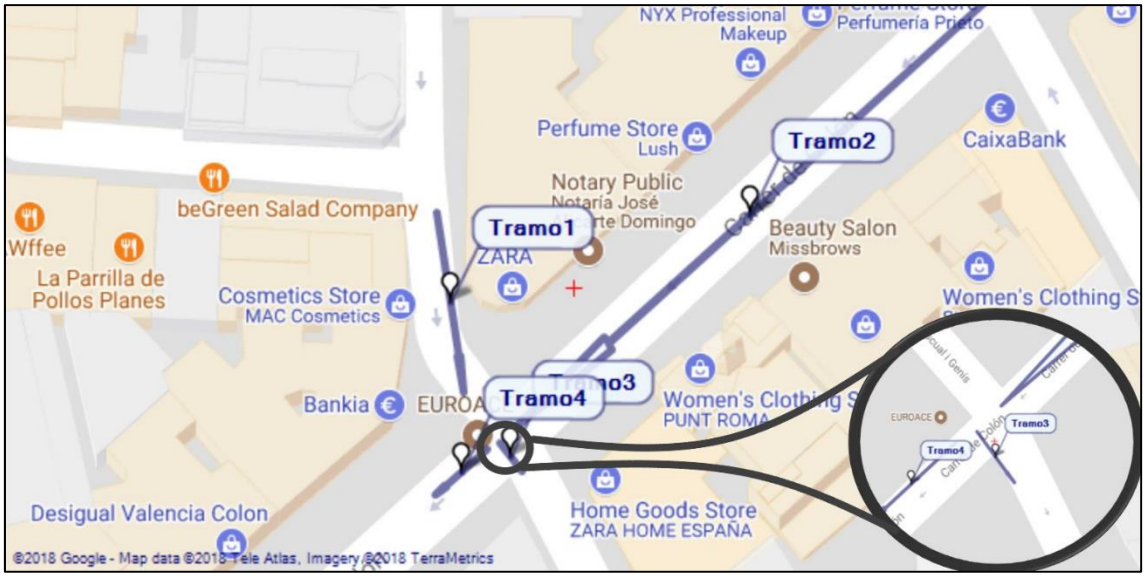
- Ubicar los Tramos que forman parte de las Zonas de Estudio.

En total existen cuatro áreas que cumplen con las características que deben tener todos los Tramos, estas son las siguientes:

- Se componen de un trayecto desde un punto inicial hasta un punto final sin influencia de otras vías o áreas.
- Tienen un único sentido de inicio a fin.
- En el punto inicial o en cualquier punto intermedio no deben tener ningún elemento regulador del tránsito (semáforos, carteles…), estos deben estar situados únicamente en el punto final.

Por lo tanto, los Tramos que cumplen con estas características son los siguientes:

Ilustración 37: Tramos de la zona.



- Con los tramos ubicados, es necesario relacionar los tramos entre sí mediante Importancias.

Esta relación originalmente se extrae de un estudio previo a realizar, en el cual se cuentan los vehículos y peatones que transitan por la zona. Haciendo posible de esta forma, extraer conclusiones exactas.

En cambio, como no se dispone de ningún estudio o valores experimentales, se van a suponer los siguientes valores de importancia.

Entradas / Salidas	Tramo 1	Tramo 2	Tramo 3	Tramo 4	TOTAL (%)
Tramo 1	0	0	50	50	100
Tramo 2	0	0	30	70	100
Tramo 3	0	0	0	0	0
Tramo 4	0	0	0	0	0
TOTAL (%)	0	0	80	120	

Tabla 3: Importancias desglosadas.

Las columnas con un total de importancia equivalente a 0, expresan las entradas del tipo Principal, es decir, aquellas que no están afectadas por ningún Tramo dentro de la Zona de Estudio sino por valores establecidos por el usuario de flujos de vehículos ajenos al área de estudio.

Al tener todas las filas con valores de 0% o 100%, se puede afirmar que los datos estipulados son correctos, otros valores serian indicativos de error dado que los porcentajes no cuadran.

Conocidos los valores de Importancias es posible llegar a las conclusiones que aporta la Tabla, las cuales son:

1. Los Tramos 1 y 2 son Entradas Principales.
2. El 50% de los vehículos del Tramo 1 y el 30% de los correspondientes al Tramo 2 se dirigen al Tramo 3.
3. El Tramo 4 recibe la mayoría (70%) de los vehículos del Tramo 2 y la mitad (50%) de los vehículos del Tramo 1.

Con los Tramos presentados y conocidos los valores de Importancias, el próximo paso trata de conocer las distintas variables que van a tener que ser incorporadas en la aplicación para llevar a cabo una correcta simulación.

Variables	Tramo 1	Tramo 2	Tramo 3	Tramo 4	Necesaria para obtener:
Coordenadas Iniciales	39°28'05.6"N 0°22'26.0"O	39°28'06.5"N 0°22'22.6"O	39°28'04.4"N 0°22'25.4"O	39°28'04.4"N 0°22'25.5"O	Distancia
Coordenadas Finales	39°28'04.5"N 0°22'25.5"O	39°28'04.5"N 0°22'25.4"O	39°28'04.1"N 0°22'25.3"O	39°28'04.2"N 0°22'25.8"O	Distancia
Número de Carriles	2	3	1	4	Capacidad
Tipo de Semáforos	Verde, Ámbar y Rojo ()	Verde, Ámbar y Rojo ()	Ámbar y Rojo ()	Verde, Ámbar y Rojo ()	Simulación
Velocidad Media	15	22	8	12	Tiempos
Flujo de Vehículos	Normal (μ,σ) (3,1)	Normal (μ,σ) (5,1)	-	-	Vehículos (Entrantes)
Flujo de Peatones	Rampa Negativa μ = 1	Normal (μ,σ) (4,1)	Rampa Negativa μ = 1	Rampa Positiva μ = 4	Peatones

Tabla 4: Valores de Interés de la Prueba

Conocidos todos los valores necesarios llega el momento de emplearlos en el simulador llevando a cabo las Etapas explicadas hasta obtener el Archivo de Texto que contiene la configuración. Este archivo contiene la siguiente información:

Ilustración 38: Contenido del archivo de configuración.

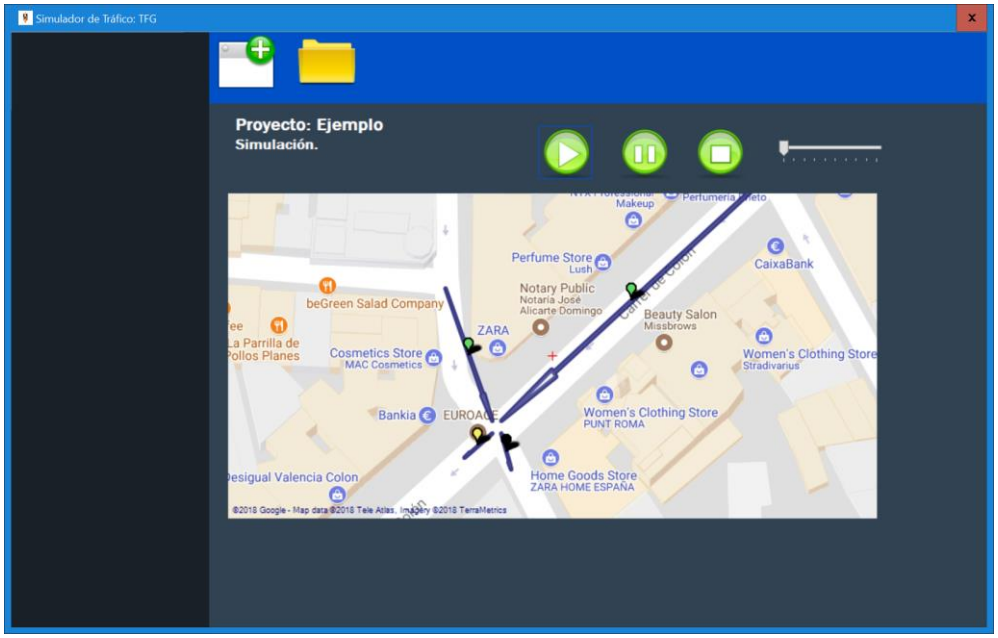
```
Ejemplo
NOMBRES
Tramo1
Tramo2
Tramo3
Tramo4
LATITUDES INICIALES
39.468222,39.468472,39.467889,39.467889
LONGITUDES INICIALES
-0.373889,-0.372944,-0.373722,-0.37375
LATITUDES FINALES
39.467917,39.467917,39.467806,39.467833
LONGITUDES FINALES
-0.37375,-0.373722,-0.373694,-0.373833
VELOCIDADES KM/H
15,22,8,12
TIPOS
1,2,3,2
CARRILES
2,3,1,3
DISTANCIAS
0.0359924299891593,0.0910334564203908,0.00954770143576387,0.00947297761366822
DENSIDAD VEHICULOS
1,3,1
1,5,1


DENSIDAD PEATONES
2,1,0
1,4,1
2,1,0
2,4,1
IMPORTANCIAS
0,0,0,0
0,0,0,0
50,30,0,0
50,70,0,0
SUMA IMPORTANCIAS
0,0,80,50
```

Una vez obtenido el archivo de configuración, es posible llevar a cabo la simulación, para ello se pulsa el botón Iniciar para que dé comienzo y se regula la velocidad de simulación al máximo (x10) hasta que haya transcurrido el tiempo, durante el cual todos los Tramos hayan estado tanto Activos como Parados varias veces.

Transcurrido ese tiempo ya es aconsejable detener la Simulación puesto que en la salida ya habrá suficientes resultados para sacar conclusiones.

Ilustración 39: Simulación.



Una vez detenida la simulación se escoge un nombre y la carpeta donde guardar el Archivo de Texto que va a contener todos los valores de iteración, Vehículos, Peatones, Tiempos Restante y Capacidades de todos los Tramos.

El Archivo de Texto se puede observar en el Anexo IV.

Una vez obtenido este Archivo de Texto es posible importar los resultados y, por tanto, se puede obtener una representación gráfica de la cantidad de vehículos en función del tiempo que ha durado la Simulación para todos los Tramos.

Las representaciones obtenidas son las siguientes:

- **Tramo 1**: Se ha obtenido una respuesta con un máximo de 8 vehículos con lo que su comportamiento entra en el margen limitado por una capacidad de 14 vehículos. Añadir que el comportamiento del Tramo se repite durante la simulación aunque nunca superando los 8 vehículos.

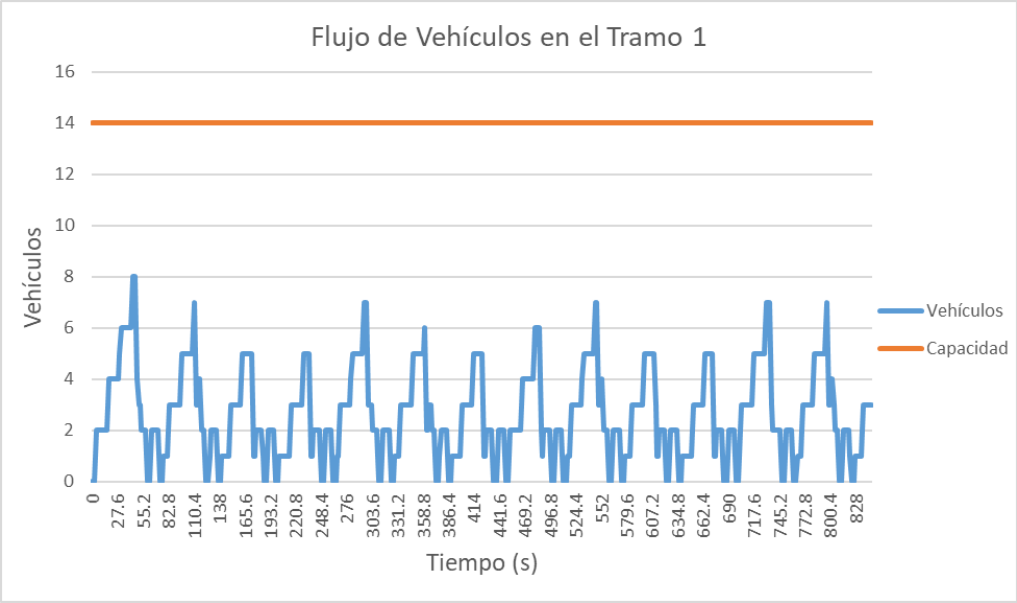


Ilustración 40: Representación gráfica Tramo 1.

- **Tramo 2**: A pesar de que esta parte de la zona de estudios recibe una gran cantidad de vehículos, se puede llevar a cabo un control que permita mantenerse alejado de la capacidad del Tramo con un máximo total de 23 vehículos. Ofreciendo además una disminución del flujo con el paso del tiempo.

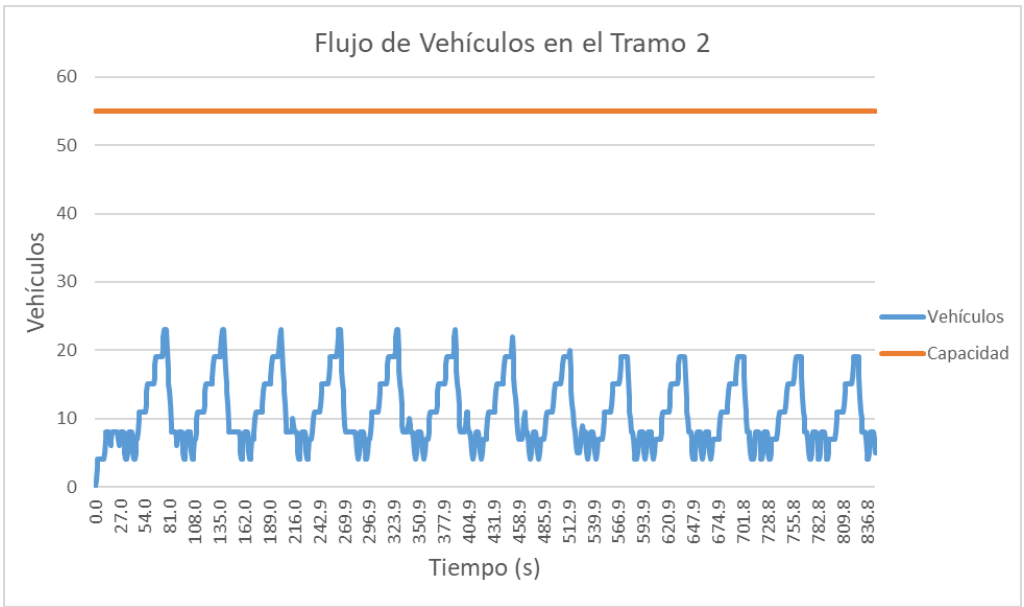


Ilustración 41: Representación gráfica Tramo 2.

- **Tramo 3**: Se trata del área con menor cantidad de vehículos dada la leve importancia que tiene. Apreciar que se supera la capacidad en varios momentos de la simulación. Hecho que puede deberse a una mala asignación de importancias o que sencillamente nos informa de que el tipo de semáforo no es el más adecuado.

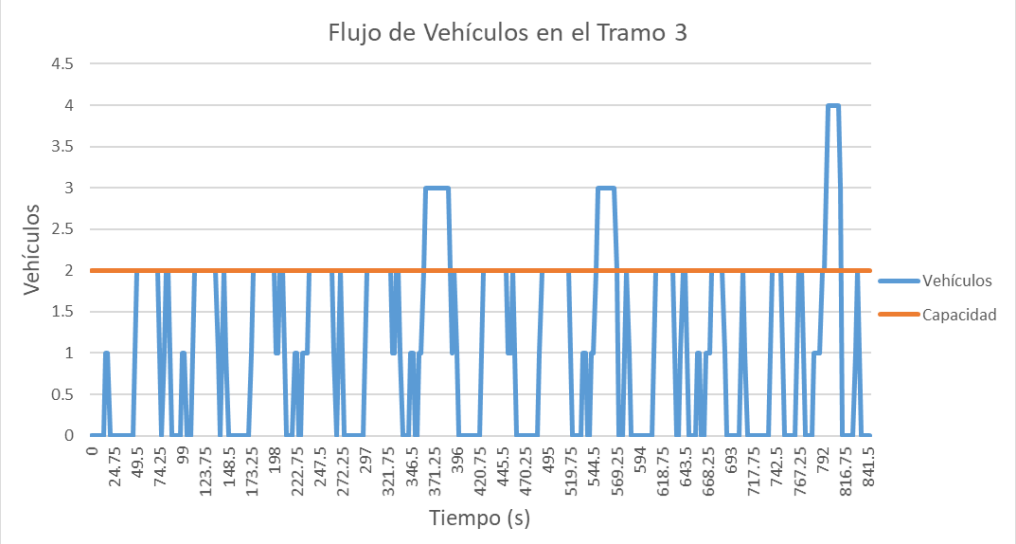
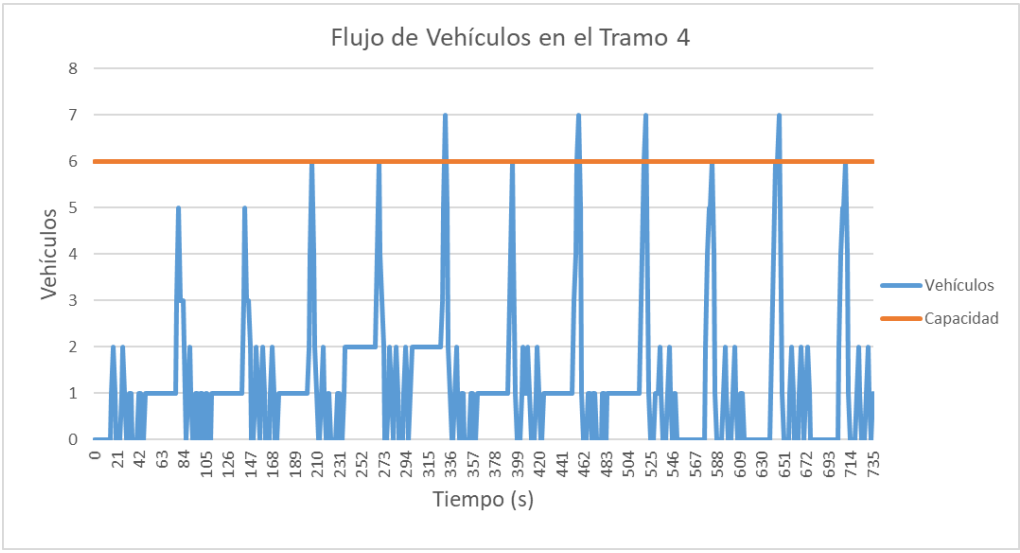


Ilustración 42: Representación gráfica Tramo 3.

- **Tramo 4**: La última de las áreas de estudio se trata de la zona con mayor valor de importancias. Hecho que produce una gran llegada de vehículos en periodos muy cortos por lo que en momentos puntuales se sobrepasa en un vehículo la capacidad. Se trata de una situación que ocurre de modo que durante la mayor parte del tiempo circulen una menor cantidad de vehículos (de 2 a 3).



-

Ilustración 43: Representación gráfica Tramo 4.

Observando las representaciones, se llega a la conclusión de que la cantidad de vehículos que transitan por los distintos Tramos está sometida bajo control dado que en momentos con altas cantidades de vehículos se cambia el estado y por ello se regulan estos valores altos, asegurando entonces un tránsito constante.

Por lo tanto, no hay peligro de saturación de Vehículos en la vía llegando a la conclusión de que los valores de tiempos medios que se pueden extraer del Archivo de Texto son valores que permiten un buen control de la Zona de Estudio.

Tiempos Medios (s)	
Activo	Parado
30.273	29.27

Tramo 1

Tiempos Medios (s)	
Activo	Parado
30	29.3

Tramo 2

Tiempos Medios (s)	
Activo	Parado
29.25	27.32

Tramo 3

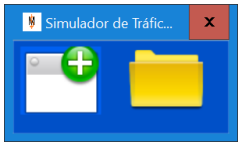
Tiempos Medios (s)	
Activo	Parado
30	30

Tramo 4

Estos valores obtenidos representan los tiempos que permiten a los Tramos no tener saturaciones. Unos tiempos que de por sí poco sirven dado que el control continuo según la cantidad de vehículos es totalmente necesario, ya que estos valores pueden variar. De todas formas, ofrecen una idea de los tiempos que se han regulado durante la simulación para obtener los resultados representados.

Manual de Usuario

La aplicación Simulador, creada para poder llevar a cabo una representación virtual del tránsito en zonas urbanas y obtener resultados que mejoren su rendimiento ofrece las siguientes opciones al iniciarse:



Ventana

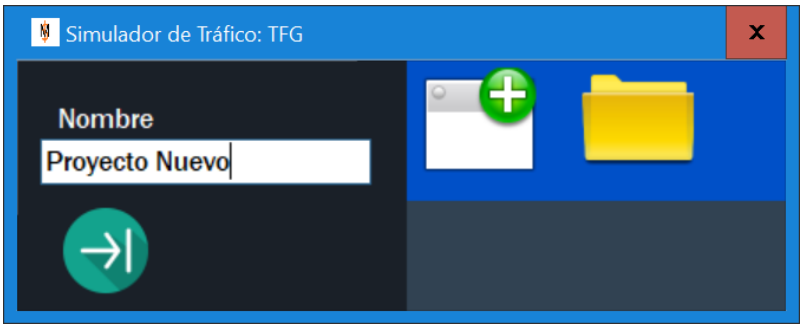
1. Pulsar el botón para iniciar un nuevo proyecto.
2. Pulsar el botón para abrir un proyecto existente.

1. Proyecto Nuevo

En caso de pulsar en nuevo proyecto, los pasos a realizar son los siguientes:

1.1. Nombre del Proyecto.

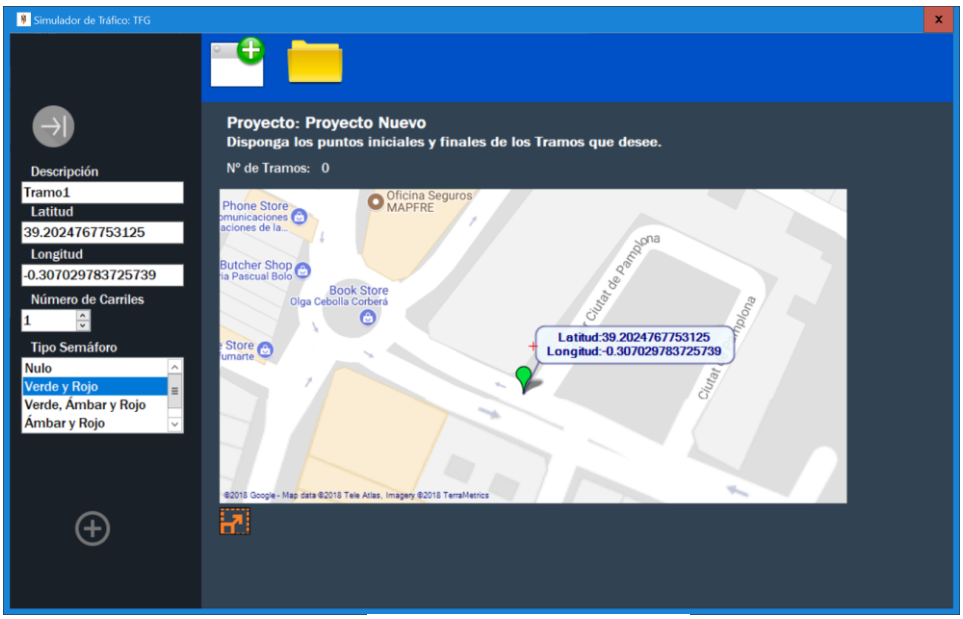
El primero de los pasos es el de escribir un nombre para el proyecto y pulsar . *(Recomendable no escribir signos de puntuación en el nombre).*



Ventana: Nombre del Proyecto.

1.2. Disposición de Tramos.

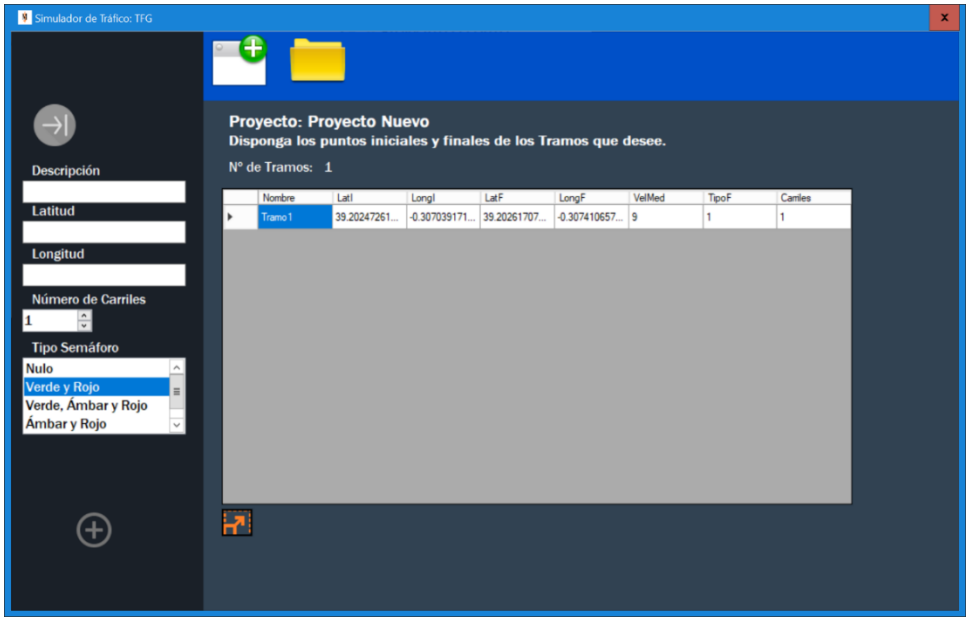
El siguiente paso es el de ubicar tantos Tramos como se desee. Para ello ha de pulsar dos veces en un punto del Mapa para ubicar el punto Inicial. Posteriormente habrá de llenar todos los datos que se pidan en el panel izquierdo de la pantalla y pulsar el botón Agregar ⊕.



Ventana: Disposición de

De igual modo, realizado el paso anterior, se debe rellenar el valor de Velocidad Media, pulsar dos veces de nuevo en el Mapa para ubicar el punto final y accionar ⊕, con lo que el Tramo queda añadido.

Para agregar tantos Tramos como se desee, se debe repetir el procedimiento pudiendo observar todos los datos guardados pulsando el botón Cambiar Vista .

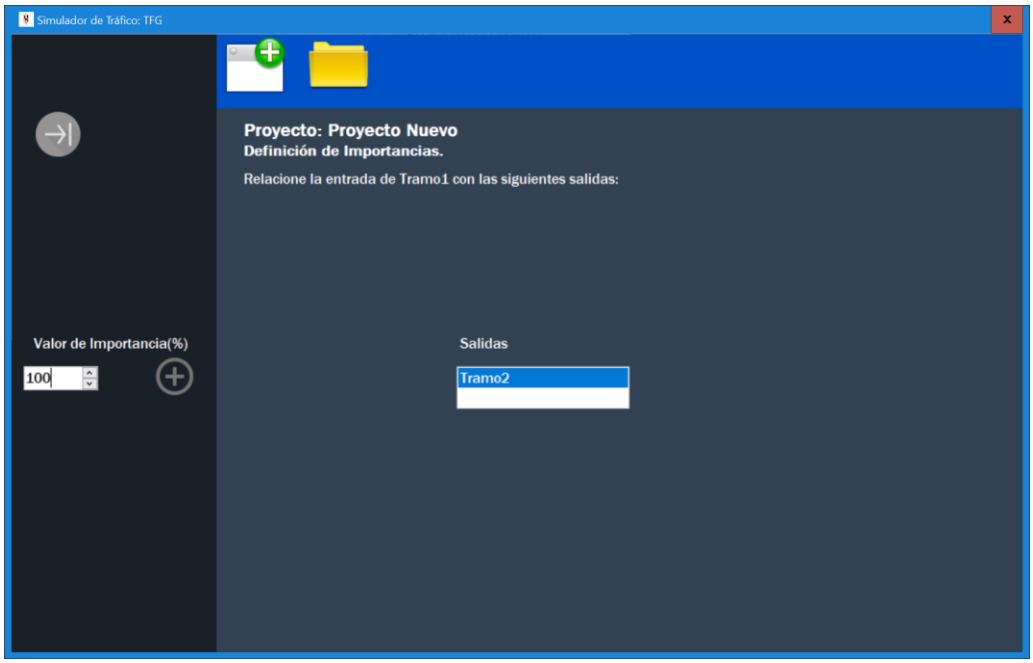


Ventana: Datos Guardados.

Dispuestos al menos dos Tramos, es posible seguir agregando más Tramos o en cambio avanzar en la aplicación pulsandoel botón Continuar .

1.3. Definición de Importancias.

En esta fase de la aplicación se deben indicar las Importancias de cada Tramo recibe a su entrada. Para indicar estas importancias se debe seleccionar el Tramo que se desee de la lista y variar el valor de la Importancia en el Seleccionador Numérico 0.

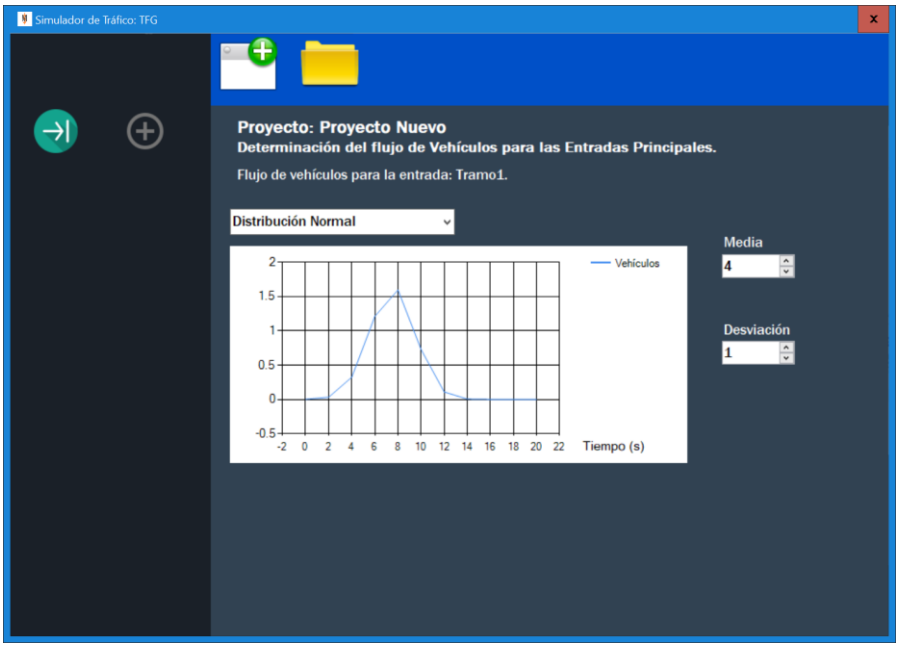


Ventana: Importancias.

Definidos los valores de cada uno de los Tramos de la lista, el siguiente paso es pulsar el botón Agregar ⊕ para que la aplicación pregunte por los demás Tramos o esta habilite el botón Continuar para poder avanzar.

1.4. Flujo Vehículos.

En esta etapa hay que definir los valores de flujos de vehículos para las Entradas Principales, por lo que para realizar esta acción es necesario seleccionar una de las expresiones disponibles en el Menú Desplegable [ ] e indicar sus valores característicos, de modo que al pulsar ⊕ se lleva a cabo la representación gráfica de los datos seleccionados.



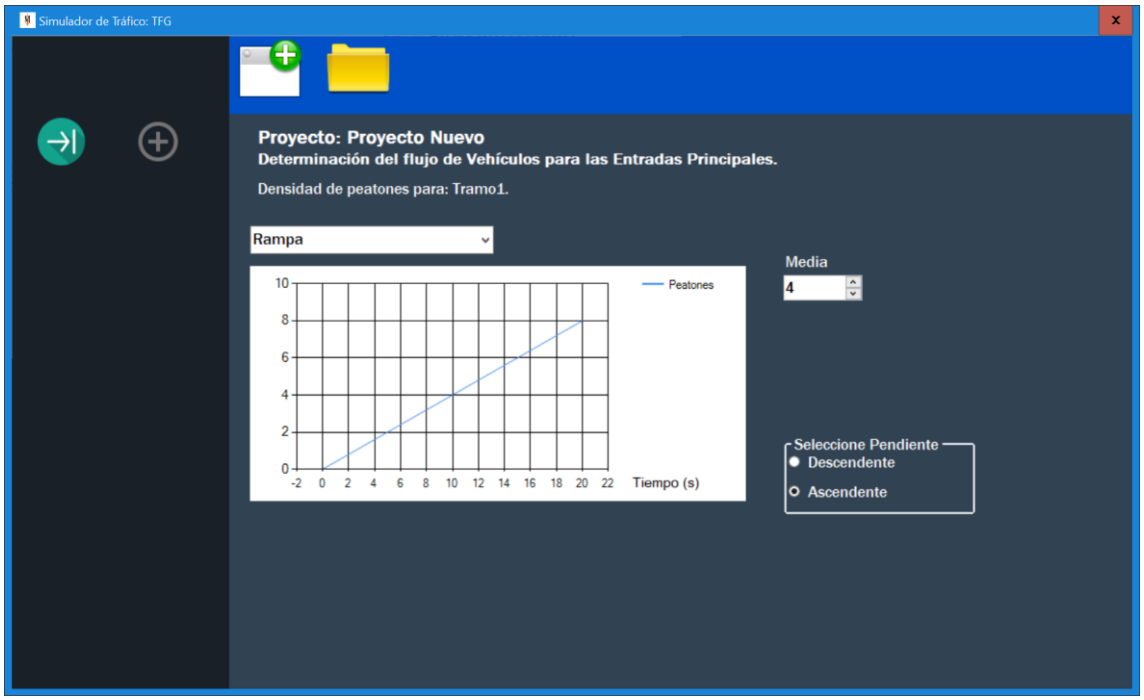
Ventana: Flujo de Vehículos.

Tenga el usuario en cuenta que la representación gráfica muestra los valores que tomará el flujo, de los cuales únicamente se tendrán en cuenta los enteros y que además, estos valores se repetiran.

Para poder definir los valores de la siguiente Entrada Principal o sencillamente para avanzar en la aplicación es necesario pulsar el botón para que el usuario pueda avanzar.

1.5. Flujo de Peatones.

Igualmente debe realizar el mismo procedimiento que en la ventana anterior, es decir, seleccionar el tipo de expresión en el Menú Desplegable y definir los valores característicos para que una vez pulsado el botón ⊕, se lleve a cabo la representación.

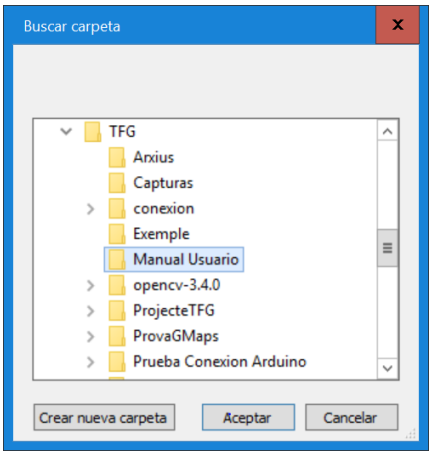


Ventana: Flujo de Peatones.

De la misma forma, para poder avanzar por la aplicación es necesario pulsar el botón Continuar .

1.6. Exportación de Configuración.

Para poder guardar la configuración aparecerá un Cuadro Emergente por pantalla tras pulsar el botón Continuar en el paso anterior. Este cuadro tiene el siguiente aspecto:

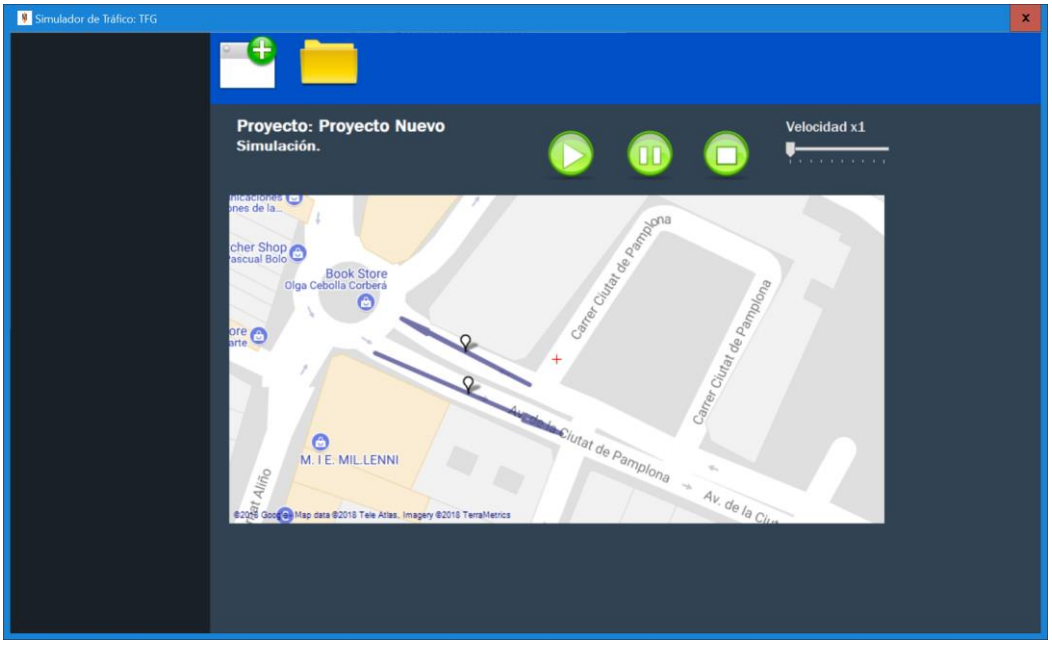


Ventana: Guardado de la Configuración.

En él será necesario seleccionar la ubicación donde guardar la configuración con el nombre del proyecto. Seleccionada la ubicación pulse aceptar.

1.7. Simulación.

Tras el paso anterior le aparecerá la siguiente ventana:



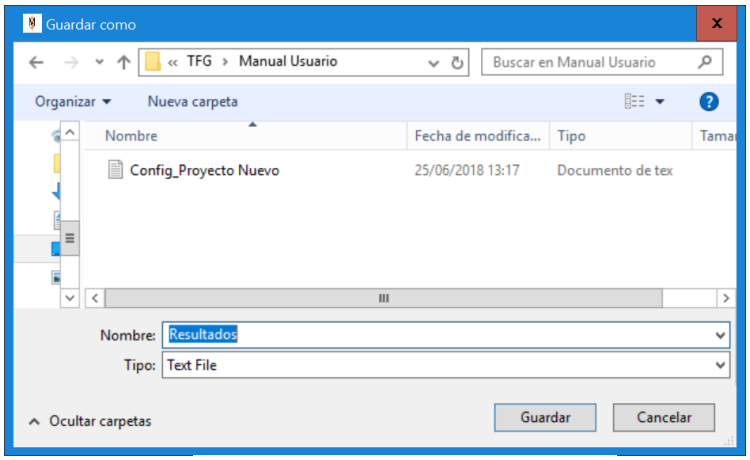
Ventana: Simulación.

Esta es la ventana de simulación donde al pulsar , esta se iniciará cambiando los estados de los Tramos. Inicialmente en cada segundo se llevará a cabo la operación de simulación, resultados de la cual se pueden observar manteniendo el ratón encima de cada marcador del mapa.

La velocidad de simulación se controla mediante el Deslizador , además de pararla pulsando el botón Detener y finalizarla con el botón Terminar .

1.8. Exportación Resultados

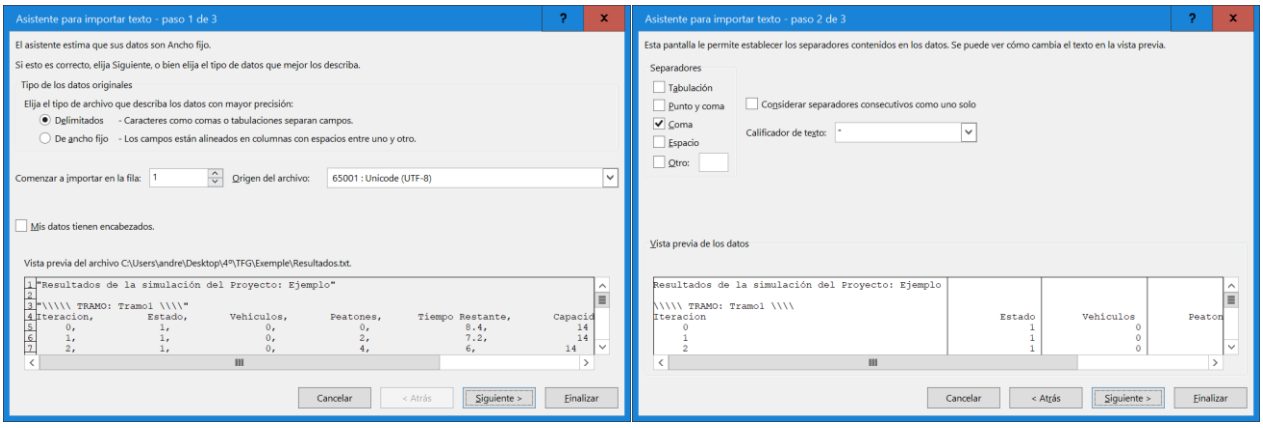
Una vez se pulsa el botón Terminar , se avanza a este paso en el cual se pide un directorio y un nombre con el que guardar el archivo con los resultados.



Ventana: Guardado de Resultados.

Una vez guardado el documento se reiniciará la ventana anterior (*Paso 1.7.*) por si se quiere volver a repetir el proceso.

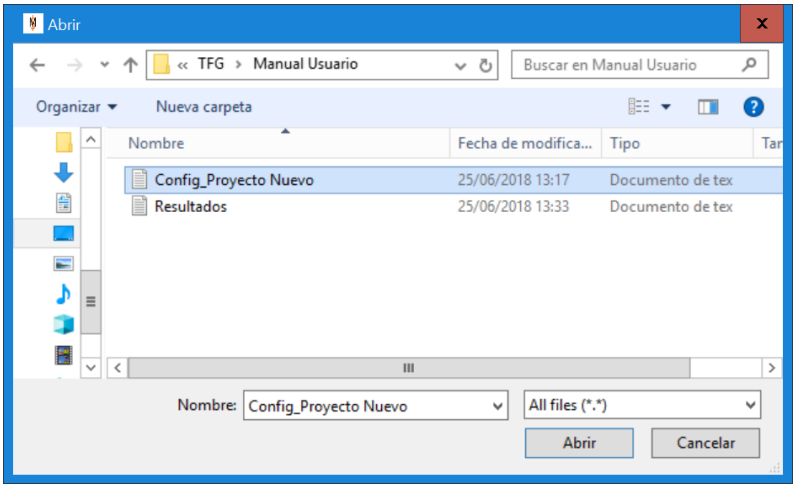
Nota: Los documentos de resultados generados por la aplicación se pueden importar desde Excel dado que los valores están definidos mediante comas. De este modo es posible integrarlos en hojas de cálculo.



Ventanas: Importación en Excel.

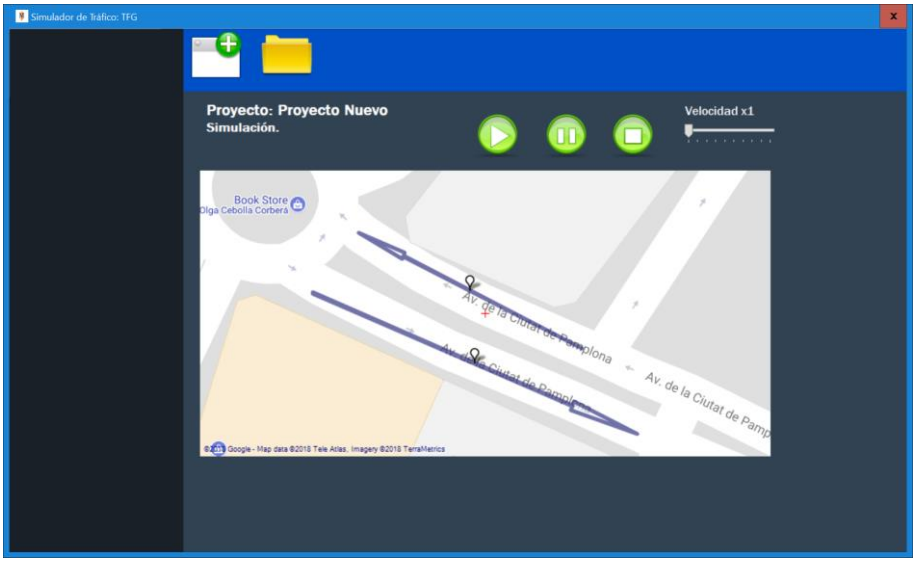
2. Cargar Proyecto

Esta es la segunda posibilidad que ofrece la aplicación al iniciarse. Si se pulsa , en este caso le aparecerá el siguiente cuadro emergente:



Ventana: Cargar Proyecto.

De modo que al seleccionar un archivo de configuración (titulados como Config_), se habilita de nuevo la ventana de simulación:



Ventana: Simulación de Proyecto

Haciendo posible la simulación de la misma forma que en el Paso 1.7. y la posterior exportación de resultados (*Paso 1.8.*)

Presupuesto

En este apartado se va a llevar a cabo un recuento de los costes de los materiales y herramientas empleadas para poder hacer posible este proyecto.

- **Mano de Obra**: En este apartado se va a tener en cuenta las horas de trabajo dedicadas y la preparación del profesional que la tiene que desempeñar.

En este caso, únicamente se va a tener en cuenta que el trabajo ha sido realizado por un ingeniero técnico, aunque en proyectos como este también sea necesaria la ayuda de otros tipos de profesionales como por ejemplo diseñadores gráficos.

Tipo de Mano de Obra	Cantidad (Horas)	Coste Unitario (€/h)	Coste Total (€)
Ingeniero Técnico	480	6.875	3300
		Subtotal	**3300**

Tabla 5: Costes de Mano de Obra.

- **Material**: Llega el turno de analizar los gastos debidas a herramientas.

Este apartado es importante debido a que el desarrollo de este proyecto sería imposible sin herramientas tecnológicas además de las tradicionales.

Material	Cantidad (ud)	Coste Unitario (€/ud)	Coste Total (€)
Ordenador Surface Pro 3	1	1100	1100
Documentación	-	-	28.6
Varios	-	-	50
		Subtotal	**1178.6**

Tabla 6: Costes de Material.

- **Licencias:** Es el apartado más importante ante todo por la Interfaz de Desarrollo que ha ofrece todas las herramientas necesarias.

Licencias	Cantidad (ud)	Coste Unitario (€/ud)	Coste Total (€)
Microsoft Office Home 2016	1	43	43
Visual Studio Profesional 2017	1	641	641
		Subtotal	**684.00**

Tabla 7: Costes de Licencias.

TOTAL: Se suman todos los costes anteriores de modo que se calcula la cantidad de dinero que supondría el desarrollo de una aplicación como esta, tal y como se ha creado.

Tipo	**Coste (€)**
Mano de Obra	3300
Material	1178.6
Licencias	684.00
TOTAL	**5162.6**

Tabla 8: Costes Totales.

El coste final es de **5162.6**€. De modo que si se quisiera comercializar sería necesario llevar a cabo el cálculo de un precio de venta que tenga en cuenta estos costes, los impuestos y un margen de beneficios.

Conclusión

Tras los resultados obtenidos se observa que el Proyecto ofrece tiempos para las distintas áreas de estudio, de modo que, con ellos, es posible mantener un buen rendimiento del tránsito en Zonas Urbanas.

En cambio, la aplicación sufre con la temporalidad debido a los problemas que surgen al tener Tramos con distintas velocidades medias. Aunque nunca afectando al Control, dado que, si se lleva a cabo un estudio sobre una determinada área obteniendo valores experimentales de la Velocidad Media y flujos de Vehículos y Peatones, es posible integrar estos valores en la aplicación y obtener tiempos correctos para mantener un buen flujo de Vehículos además de evitar problemas en el tránsito.

Con todos estos tiempos que se pueden extraer, es recomendable emplearlos en un sistema del control del tránsito real que sea capaz de poder alternar los estados de la vía pública con los tiempos obtenidos, además de tener en cuenta la lectura de los contaminantes y de flujos de vehículos con el fin de obtener nuevos tiempos en función de estos valores

También se podría informar a los conductores de puntos de la vía pública con saturaciones en el tránsito o de llevar a cabo otras acciones como reducir la velocidad o cerrar accesos a vehículos como acciones complementarias de un sistema completo del control del tránsito que tiene como posible inicio este proyecto.

Medidas que en general tienen como parte fundamental, los tiempos de alternancia de los semáforos, los cuales se pueden calcular mediante la aplicación desarrollada.

Bibliografía

- ## Estudios, Informes y Artículos sobre la contaminación.

18 GRANDES CIUDADES ESPAÑOLAS SUPERAN LOS NIVELES DE CONTAMINACIÓN ATMOSFÉRICA (CADENA SER).

http://cadenaser.com/ser/2017/11/20/ciencia/1511176298_716529.html

EVALUACIÓN CALIDAD DEL AIRE 2016 (Ministerio de Medio Ambiente).

http://www.mapama.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/informeevaluacioncalidadaireespana2016_tcm30-431898.pdf

INFORME CALIDAD DEL AIRE 2016 (Ecologistas en Acción).

https://www.ecologistasenaccion.org/IMG/pdf/informe-calidad-aire-2016.pdf

LA CONTAMINACIÓN HA MATADO A 93.000 PERSONAS EN ESPAÑA EN UNA DÉCADA (EL PAIS).

https://elpais.com/elpais/2018/06/21/ciencia/1529592814_225910.html

LA CONTAMINACIÓN ATMOSFÉRICA EMPEORÓ EN 2017 EN TODA ESPAÑA (PUBLICO.ES)

http://www.publico.es/sociedad/calidad-aire-contaminacion-atmosferica-empeoro-2017-espana.html

- ## Documentación y otras referencias relacionadas con el desarrollo.

MICROSOFT DOCS (elementos interfaz de desarrollo).

https://docs.microsoft.com/es-es/

GREATMAPS (servicio de mapas integrado).

https://greatmaps.codeplex.com/SourceControl/changeset/68b2effa78c1

ICON-ICONS (página web donde conseguir imágenes libres).

https://icon-icons.com/

La inteligencia del enjambre (EL PAIS).

https://elpais.com/sociedad/2014/02/13/actualidad/1392310892_460376.html

Anexos

Anexo I: Código Clase Formulario

```
1. using System;
2. using System.IO; //Para poder crear y leer archivos
3. using System.Collections.Generic;
4. using System.ComponentModel;
5. using System.Data; //Permite trabajar con Bases de Datos
6. using System.Drawing;
7. using System.Linq;//Permite extraer vectores a partir de cadenas
8. using System.Text;
9. using System.Threading.Tasks;
10.      using System.Windows.Forms;
11.      using System.Windows.Forms.DataVisualization.Charting; //Permite la creación
   de gráficas
12.      using GMap.NET;//Activación paquete expansión del Mapa
13.      using GMap.NET.MapProviders; //Activación proveedores
14.      using GMap.NET.WindowsForms;
15.      using GMap.NET.WindowsForms.Markers; //Permite crear Marcadores
16.
17.
18.      namespace ProjecteTFG
19.      {
20.          public partial class Form1 : Form //Clase de la Ventana del Proyecto
21.          {
22.              //DEFINICIÓN E INICIALIZACIÓN DE LAS VARIABLES GENERALES DEL
   PROYECTO
23.
24.
25.              GMarkerGoogle marker; //Marcador del mapa
26.              GMapOverlay markerOverlay, markerOverlay1; //Definición del Elemento
   Mapa
27.              DataTable dtSimTram,dtSimRel, Sim, Resultados; //Definición de las
   Tablas de Datos
28.              Algoritmo Simulador = new Algoritmo(); //Instancia con la que
   utilizar las funciones de la clase algoritmo
29.
30.              //Definición variables enteras
31.              int nSemaforos, faseTramo, nTramo, iteracion, MinAmbar = 2,
   expandir = 0, CiclosR;
32.              int filaSeleccionada = 0;
33.              bool Agregar = false;
34.              //Definicion variables Double
35.              double fase = 0.0, Tiempo, LatInicial = 39.220459, LngInicial = -
   0.255306; //Coordenadas escogidas al azar
36.
37.              //Definición Listas
38.              List<String> Intermedios = new List<String>(),
   Salidas = new List<String>(), Distribuciones = new List<string>();
39.
40.              //Definición vectores Double
```

```
41.             double[] Importancias, Configuracion = { 0.0, 0.0, 0.0 },
   EntradasPrincipales = new double[30000];
42.
43.             //Definición Cadenas
44.             string NombreProyecto, importancias, dir, Estados, Vehiculos,
   Peatones, TiemposEstado;
45.             string[] Capacidades;
46.
47.
48.             //Ruta automatizada
49.             PointLatLng inicio;
50.             PointLatLng final;
51.
52.
53.
54.             public Form1()
55.             {
56.                 InitializeComponent();
57.             }
58.
59.             private void textBox1_TextChanged(object sender, EventArgs e)
60.             {
61.                 //Si varia el texto del Cuadro de Texto, se habilita el botón
   Agregar
62.                 btnAgregar.Enabled = true;
63.             }
64.
65.             private void gMapControl1_Load(object sender, EventArgs e)
66.             {
67.
68.             }
69.
70.             private void Form1_Load_1(object sender, EventArgs e)
71.             {
72.
73.                 Temporizador.Stop(); //Inicialmente el Temporizador está parado
74.
75.                 dtSimTram = new DataTable(); //Tabla de Datos 1
76.                 dtSimTram.Columns.Add(new DataColumn("Nombre", typeof(string)));
    //0 Nombre de los Tramos
77.                 dtSimTram.Columns.Add(new DataColumn("LatI", typeof(double)));
   //1 Latitud Punto Inicial
78.                 dtSimTram.Columns.Add(new DataColumn("LongI", typeof(double)));
   //2 Longitud Punto Inicial
79.                 dtSimTram.Columns.Add(new DataColumn("LatF", typeof(double)));
   //3 Latitud Punto Final
80.                 dtSimTram.Columns.Add(new DataColumn("LongF", typeof(double)));
   //4 Longitud Punto Final
81.                 dtSimTram.Columns.Add(new DataColumn("VelMed", typeof(double)));
      //5 Velocidad Media de los Tramos
82.                 dtSimTram.Columns.Add(new DataColumn("TipoF", typeof(int)));    /
   /6 Tipo Semáforo
```

```
83.                 dtSimTram.Columns.Add(new DataColumn("Carriles", typeof(int)));
    //7 Número de Carriles
84.                 dtSimTram.Columns.Add(new DataColumn("Distancia", typeof(double)
    )); //8 Distancia entre P Inicial y Final
85.                 dtSimTram.Columns.Add(new DataColumn("Vehiculos", typeof(string)
    )); //9 Modelo del Flujo de Vehículos
86.                 dtSimTram.Columns.Add(new DataColumn("VACIO2", typeof(bool))); /
    /10
87.                 dtSimTram.Columns.Add(new DataColumn("PeatonesF", typeof(string)
    ));  //11 Modelo Flujo de Peatones
88.
89.                 dtSimRel = new DataTable(); //Tabla de Datos 2
90.                 dtSimRel.Columns.Add(new DataColumn("Nombre", typeof(string)));
    //Nombre de cada Tramo
91.                 dtSimRel.Columns.Add(new DataColumn("Importancia", typeof(string
    ))); //Importancias de cada Tramo
92.                 dtSimRel.Columns.Add(new DataColumn("SumaImportancia", typeof(do
    uble))); //Suma de Importancias
93.
94.                 Sim = new DataTable(); //Tabla de Datos 3
95.                 Sim.Columns.Add(new DataColumn("Estado", typeof(bool)));//Estado
    Actual (F->Rojo, V->Verde)  0
96.                 Sim.Columns.Add(new DataColumn("CiclosA", typeof(int)));//Ciclos
    en estado actual    1
97.                 Sim.Columns.Add(new DataColumn("CiclosR", typeof(int)));//Ciclos
    Restantes   2
98.                 Sim.Columns.Add(new DataColumn("VectorV", typeof(string)));//Vec
    tor de ciclos de vehiculos  3
99.                 Sim.Columns.Add(new DataColumn("VehiculosT", typeof(int)));//Veh
    ículos en el tramo 4
100.                Sim.Columns.Add(new DataColumn("PeatonesT", typeof(int)));//Peat
    ones en el tramo  5
101.                Sim.Columns.Add(new DataColumn("VehiculosEnV", typeof(int)));//V
    ehiculos al final del ultimo periodo Verde   6
102.                Sim.Columns.Add(new DataColumn("VehiculosEnR", typeof(int)));//V
    ehiculos al final del ultimo periodo Rojo 7
103.                Sim.Columns.Add(new DataColumn("UltimoV", typeof(int)));//Ultimo
    vehículo del vector 8
104.                Sim.Columns.Add(new DataColumn("TiempoR", typeof(double)));//Tie
    mpo Restante para avanzar ciclo 9
105.                Sim.Columns.Add(new DataColumn("Tiempo", typeof(double)));//Tiem
    po para avanzar ciclo 10
106.                Sim.Columns.Add(new DataColumn("Alternancia", typeof(bool)));//A
    lternancia para ambar  11
107.                Sim.Columns.Add(new DataColumn("Iteracion", typeof(int)));//Iter
    acion de cada estado 12
108.                Sim.Columns.Add(new DataColumn("Interseccion", typeof(int)));//T
    ramo a activado en intersección 13
109.
110.                Resultados = new DataTable(); //Tabla de Datos 4
111.                Resultados.Columns.Add(new DataColumn("Iteracion", typeof(int)))
    ;//Iteracion de la simulación
```

```
112.                 Resultados.Columns.Add(new DataColumn("Estados", typeof(string))
     );//Estado Actual (F->Rojo, V->Verde)
113.                 Resultados.Columns.Add(new DataColumn("Vehiculos", typeof(string
     )));//Vehiculos
114.                 Resultados.Columns.Add(new DataColumn("Peatones", typeof(string)
     ));//Peatones
115.                 Resultados.Columns.Add(new DataColumn("Capacidad", typeof(string
     )));//Capacidad Vehículos
116.                 Resultados.Columns.Add(new DataColumn("Tiempo", typeof(string)))
     ;//Tiempo total en este estado
117.
118.                 //Definición de los valores de las listas
119.                 List<String> Semaforos_TipoO = new List<String>(); //Lista para
     los semáforos
120.                 Semaforos_TipoO.Add("Nulo");
121.                 Semaforos_TipoO.Add("Verde y Rojo");
122.                 Semaforos_TipoO.Add("Verde, Ámbar y Rojo");
123.                 Semaforos_TipoO.Add("Ámbar y Rojo");
124.                 Semaforos_TipoO.Add("Ámbar");
125.                 tipoOut.DataSource = Semaforos_TipoO;
126.
127.
128.                 List<String> Distribuciones = new List<string>(); //Lista para
     las distribuciones
129.                 Distribuciones.Add(" ");
130.                 Distribuciones.Add("Distribución Normal");
131.                 Distribuciones.Add("Rampa");
132.                 Distribuciones.Add("Distribución Constante");
133.                 comboDis.DataSource = Distribuciones;
134.
135.
136.                 //Configuración del Mapa
137.                 gMapControl1.DragButton = MouseButtons.Left; //Click izquierdo
     para mover el mapa
138.                 gMapControl1.CanDragMap = true;//Permite mover el mapa
139.                 gMapControl1.MapProvider = GMapProviders.GoogleMap; //El
     proveedor de cartografía es Google Maps
140.                 gMapControl1.Position = new PointLatLng(LatInicial,
     LngInicial); //Posición donde se enfoca inicialmente
141.                 gMapControl1.MinZoom = 0; //Zoom mínimo
142.                 gMapControl1.MaxZoom = 24; //Zoom máximo
143.                 gMapControl1.Zoom = 18; //Zoom que tiene el Mapa al iniciar la
     aplicación
144.                 //gMapControl1.AutoScroll = true; //Aplicación de Zoom
     automática
145.
146.
147.                 //Creación de la capa de marcadores con la que se va a trabajar
     posteriormente
148.                 markerOverlay = new GMapOverlay("Marcador");
149.                 marker = new GMarkerGoogle(new PointLatLng(0.0, 0.0),
     GMarkerGoogleType.green);
150.                 markerOverlay.Markers.Add(marker); //Agregamos al mapa
```

```
151.
152.                 //agregamos un toolip de texto a los marcadores.
153.                 marker.ToolTipMode = MarkerTooltipMode.Always;
154.
155.                 //agregar el mapa y el marcador al mapa de control
156.                 gMapControl1.Overlays.Add(markerOverlay);
157.
158.                 markerOverlay1 = new GMapOverlay("Marcadores");
159.                 gMapControl1.Overlays.Add(markerOverlay1);
160.                 //Tamaño original del formulario y panel izquierdo
161.                 panel2.Size = new Size(0, 163);
162.                 this.Size = new Size(200, 118);
163.
164.             }
165.
166.             private void gMapControl1_MouseDoubleClick(object sender,
    MouseEventArgs e)
167.             {
168.                 if (fase < 1.3)
169.                 {
170.                     //obtención de lat y long donde se pulsó
171.                     gMapControl1.Overlays[1].Clear();
172.                     double lat = gMapControl1.FromLocalToLatLng(e.X, e.Y).Lat;
173.                     double lng = gMapControl1.FromLocalToLatLng(e.X, e.Y).Lng;
174.                     //CrearDireccionTrazarRuta(lat, lng);
175.                     //se posicionan en el txt de latitud y longitud
176.                     txtlatitud.Text = lat.ToString();
177.                     txtlongitud.Text = lng.ToString();
178.                     //Se crea el marcador para moverlo al lugar indicado
179.                     marker = new GMarkerGoogle(new PointLatLng(lat, lng),
    GMarkerGoogleType.green);
180.                     markerOverlay1.Markers.Add(marker); //Agregamos al mapa
181.                                                         //Se agrega el mensaje
    al marcador (tooltip)
182.                     marker.ToolTipText = string.Format("Latitud:{0} \nLongitud:{
    1}", lat, lng);
183.                     marker.ToolTipMode = MarkerTooltipMode.Always;
184.                     gMapControl1.Overlays.Add(markerOverlay1);
185.                 }
186.
187.             }
188.
189.
190.
191.             private void btnAgregar_Click(object sender, EventArgs e)
192.             {
193.                 double LtI, LtF, LgI, LgF;
194.                 //Al pulsar el botón agregar se realizan una acción en función
    de la fase (etapa) del programa
195.                 switch (fase)
196.                 {
197.                     case 1.1:
```

```
198.                         if (faseTramo == 0) //Si se trata del punto inicial, se
    guardan sus valores y se pide la velocidad media
199.                         {
200.                             if (txtdescripcion.Text != "" && txtlongitud.Text !=
     "" && txtlongitud.Text !=  "")
201.                             {
202.                                 faseTramo = 1;
203.                                 dtSimTram.Rows.Add(txtdescripcion.Text,
    Convert.ToDouble(txtlatitud.Text),
    Convert.ToDouble(txtlongitud.Text), 0.0, 0.0, 0.0, tipoOut.SelectedIndex,
    numerico.Value, 0.0, "", false, "");
204.                                 //Se resetean los elementos de la ventana
205.                                 txtdescripcion.Enabled = false;
206.                                 txtVelMed.Visible = true;
207.                                 txtVelMed.Text = "0.0";
208.                                 label4.Visible = true;
209.                                 markerOverlay.Markers.Add(marker);
210.                                 gMapControl1.Overlays.Add(markerOverlay);
211.                             }
212.                             else { }
213.
214.                         } else if(Convert.ToDouble(txtVelMed.Text)>0.0)//Si se
    trata de la velocidad final, se guardan todos los valores (especialmente las
    coordenadas finales)
215.                         {
216.                             LtI = Convert.ToDouble(dataGridView1.Rows[dataGridVi
    ew1.Rows.Count - 1].Cells[1].Value);
217.                             LgI = Convert.ToDouble(dataGridView1.Rows[dataGridVi
    ew1.Rows.Count - 1].Cells[2].Value);
218.                             LtF = Convert.ToDouble(txtlatitud.Text);
219.                             LgF = Convert.ToDouble(txtlongitud.Text);
220.                             if (Obtener_Distancia(LtI, LgI, LtF,
    LgF) * 1000.0 > 0.0)
221.                             {
222.                                 dataGridView1.Rows[dataGridView1.Rows.Count - 1]
    .Cells[3].Value = Convert.ToDouble(txtlatitud.Text); //Latitud Final
223.                                 dataGridView1.Rows[dataGridView1.Rows.Count - 1]
    .Cells[4].Value = Convert.ToDouble(txtlongitud.Text); //Longitud Final
224.                                 dataGridView1.Rows[dataGridView1.Rows.Count - 1]
    .Cells[8].Value = Obtener_Distancia(Convert.ToDouble(dataGridView1.Rows[dataGridV
    iew1.Rows.Count - 1].Cells[1].Value),
    Convert.ToDouble(dataGridView1.Rows[dataGridView1.Rows.Count - 1].Cells[2].Value)
    ,
    Convert.ToDouble(dataGridView1.Rows[dataGridView1.Rows.Count - 1].Cells[3].Value)
    ,
    Convert.ToDouble(dataGridView1.Rows[dataGridView1.Rows.Count - 1].Cells[4].Value)
    ); //Longitud Final
225.                                 dataGridView1.Rows[dataGridView1.Rows.Count - 1]
    .Cells[5].Value = Convert.ToDouble(txtVelMed.Text); //Velocidad Media
226.                                 dataGridView1.Rows[dataGridView1.Rows.Count - 1]
    .Cells[6].Value = tipoOut.SelectedIndex; //Reguardado tipo Semáforos
227.                                 dataGridView1.Rows[dataGridView1.Rows.Count - 1]
    .Cells[7].Value = numerico.Value; //Reguardado numero Carriles
```

```
228.                                 faseTramo = 0;
229.                                 LCuenta.Text = dtSimTram.Rows.Count.ToString();
230.                                 //Se resetean los elementos de la ventana
231.                                 txtdescripcion.Enabled = true;
232.                                 nTramo++;
233.                                 txtdescripcion.Text = "";
234.                                 txtlatitud.Text = "";
235.                                 txtlongitud.Text = "";
236.                                 txtVelMed.Visible = false;
237.                                 label4.Visible = false;
238.                                 markerOverlay.Markers.Add(marker);
239.                                 gMapControl1.Overlays.Add(markerOverlay);
240.                             }
241.
242.                         }
243.
244.                         nSemaforos = dataGridView1.Rows.Count; //Se actualiza la
    cantidad de Tramos
245.                         if(nTramo >=2 && faseTramo == 0)
246.                         {
247.                             btnOk.Enabled = true; //Si hay dos o más tramos, se
    permite continuar a la siguiente fase
248.                         }
249.
250.                         break;
251.                     case 1.2: //Fase de selección de importancias
252.
253.                         double SI = SumaImportancias(Importancias,
    faseTramo); //Obtiene la suma de importancias del Tramo: fase Tramo
254.                         //Se añade una nueva fila con
255.                         dtSimRel.Rows.Add(dataGridView1.Rows[faseTramo].Cells[0]
    .Value.ToString(), importancias, SI);
256.                         //Valores del vector importancias a 0
257.                         limpiarArray(); //Valores del vector importancias a 0
258.                         //Se actualiza los elementos de la lista
259.                         Salidas.Clear();
260.                         listaSalidas.Items.Clear();
261.                         Agregar = true;
262.                         numerico.Value = 0;
263.                             //Se llena la lista con todos los tramos
264.                             for (int filas = 0; filas < dataGridView1.Rows.Count
    ; filas++)
265.                             {
266.                                 Salidas.Add(Convert.ToString(dataGridView1.Rows[
    filas].Cells[0].Value));
267.                                 listaSalidas.Items.Add(Salidas[filas].ToString()
    );
268.                             }
269.                             faseTramo++;
270.                 //Si ya se han todas las importancias de todos los Tramos, se
    preparan los elementos de la siguiente fase
271.                         if (faseTramo == dtSimTram.Rows.Count)
272.                             {
```

```
273.                                 listaSalidas.Visible = false;
274.                                     btnAgregar.Enabled = true;
275.                                     label9.Visible = false;
276.                                     faseTramo = -1;
277.                                     fase = 1.3;
278.                                 //Configuración de la gráfica a utilizar en la
    siguiente fase
279.                                     Graf.Series.Clear(); //Reseteo valores
280.                                     Graf.Series.Add("Vehículos"); //Nueva serie de
    valores
281.                                     Graf.Series["Vehículos"].ChartType = SeriesChart
    Type.Line;//Tipo de representación
282.                                 //Actualización elementos, si estos desaparecen o no
283.                                     btnAgregar.Visible = false;
284.                                     label6.Visible = false;
285.                                     numerico.Visible = false;
286.                                     btnOk.Enabled = true;
287.                                 }
288.                                 else
289.                                 {
290.                                 //Si el programa todavía tiene que definir
    importancias, se elimina de la lista
291.                                 //el Tramo por el que se está preguntando.
292.                                     listaSalidas.Items.RemoveAt(faseTramo);

293.                                     LInstruccion2.Text = string.Format("Relacione la
    entrada de {0} con las siguientes salidas",
    dataGridView1.Rows[faseTramo].Cells[0].Value.ToString());
294.
295.                                 }
296.
297.                             break;
298.
299.
300.                         case 1.3: //Fase flujo de vehículos
301.
302.                             switch (comboDis.SelectedIndex)
303.                             {//En función del valor seleccionado de la lista
    desplegable
304.                                 case 1: //Para una distribución normal
305.
306.                                     if(numerico3.Value > 0 && numerico2.Value > 0)
307.                                     {
308.                                         Graf.Series["Vehículos"].Points.Clear();
309.                                         //Función que calcula los valores a
    representar
310.                                         int[] Y = DistribucionNormal(Convert.ToDoubl
    e(numerico3.Value), Convert.ToDouble(numerico2.Value), 10);
311.                                         for(int i = 1;i <= Y.Length; i++)
312.                                         {
313.                                             Graf.Series["Vehículos"].Points.AddXY(Ma
    th.Round(Simulador.CalcularTiempo(Convert.ToDouble(dtSimTram.Rows[faseTramo][5]))
    *(i-1),2),Y[i-1]);
```

```
314.                                     }
315.                                     //Se calcula la gráfica y se guarda la
    configuración en la Tabla de Datos 1.
316.                                     dtSimTram.Rows[faseTramo][9]=  Convert.ToDou
    ble(comboDis.SelectedIndex).ToString() + "," + Convert.ToDouble(numerico3.Value).
    ToString() + "," + Convert.ToDouble(numerico2.Value).ToString();
317.                                     btnOk.Enabled = true;
318.
319.                                 }
320.                                 break;
321.                             case 2: //Para una distribución en
    rampa
322.                                 if((Asc.Checked || Desc.Checked) && numerico3.Va
    lue >0)
323.                                 {
324.                                     Graf.Series["Vehículos"].Points.Clear();
325.                                     //Función que calcula los valores a
    representar
326.                                     int[] Y = DistribucionEnRampa(Convert.ToDoub
    le(numerico3.Value),10,Asc.Checked);
327.                                     for (int i = 1; i <= Y.Length; i++)
328.                                     {
329.                                         Graf.Series["Vehículos"].Points.AddXY(Ma
    th.Round(Simulador.CalcularTiempo(Convert.ToDouble(dtSimTram.Rows[faseTramo][5]))
     * (i - 1), 2), Y[i-1]);
330.                                     }
331.                                     //Se calcula la gráfica y se guarda la
    configuración en la Tabla de Datos 1.
332.                                     dtSimTram.Rows[faseTramo][9] = Convert.ToDou
    ble(comboDis.SelectedIndex).ToString() + "," + Convert.ToDouble(numerico3.Value).
    ToString() + "," + Convert.ToDouble(numerico2.Value).ToString();
333.
334.                                     btnOk.Enabled = true;
335.
336.                                     //Graf.Series["Vehículos"].Points.AddXY(Y.Le
    ngth, Y[Y.Length - 1]);
337.                                 }
338.                                 break;
339.
340.                             case 3: //Para una distribución
    constante
341.                                 if (numerico3.Value > 0)
342.                                 {
343.                                     Graf.Series["Vehículos"].Points.Clear();
344.                                     //Función que calcula los valores a
    representar
345.                                     int[] Y = DistribucionConstante(Convert.ToDo
    uble(numerico3.Value), 10);
346.                                     for (int i = 1; i <= Y.Length; i++)
347.                                     {
348.                                         Graf.Series["Vehículos"].Points.AddXY(Ma
    th.Round(Simulador.CalcularTiempo(Convert.ToDouble(dtSimTram.Rows[faseTramo][5]))
     * (i - 1), 2), Y[i - 1]);
```

```
349.                                    }
350.                                    //Se calcula la gráfica y se guarda la
    configuración en la Tabla de Datos 1.
351.                                    dtSimTram.Rows[faseTramo][9] = Convert.ToDou
    ble(comboDis.SelectedIndex).ToString() + "," + Convert.ToDouble(numerico3.Value).
    ToString() + "," + Convert.ToDouble(numerico2.Value).ToString();
352.
353.                                    btnOk.Enabled = true;

354.                                }
355.                                break;
356.                            default: break;
357.                        }
358.                        break;
359.                    case 1.4://Fase flujo de vehículos
360.
361.                        switch (comboDis.SelectedIndex)
362.                        {//En función del valor seleccionado de la lista
    desplegable
363.                            case 1://Para una distribución normal
364.
365.                                if (numerico3.Value > 0 && numerico2.Value > 0)
366.                                {
367.                                    Graf.Series["Peatones"].Points.Clear();
368.                                    //Función que calcula los valores a
    representar
369.                                    int[] Y = DistribucionNormal(Convert.ToDoubl
    e(numerico3.Value), Convert.ToDouble(numerico2.Value), 10);
370.                                    for (int i = 1; i <= Y.Length; i++)
371.                                    {
372.                                        Graf.Series["Peatones"].Points.AddXY(Mat
    h.Round(Simulador.CalcularTiempo(Convert.ToDouble(dtSimTram.Rows[faseTramo][5]))
    * (i - 1), 2), Y[i - 1]);
373.                                    }
374.                                    btnOk.Enabled = true;
375.                                    // Se calcula la gráfica y se guarda la
    configuración en la Tabla de Datos 1.
376.                                    dtSimTram.Rows[faseTramo][11] = Convert.ToDo
    uble(comboDis.SelectedIndex).ToString() + "," + Convert.ToDouble(numerico3.Value)
    .ToString() + "," + Convert.ToDouble(numerico2.Value).ToString();
377.
378.                                }
379.                                break;
380.                            case 2://Para una distribución en rampa
381.                                if ((Asc.Checked || Desc.Checked) && numerico3.V
    alue > 0)
382.                                {
383.                                    Graf.Series["Peatones"].Points.Clear();
384.                                    //Función que calcula los valores a
    representar
385.                                    int[] Y = DistribucionEnRampa(Convert.ToDoub
    le(numerico3.Value), 10, Asc.Checked);
386.                                    for (int i = 1; i <= Y.Length; i++)
```

```
387.                                         {
388.                                             Graf.Series["Peatones"].Points.AddXY(Mat
    h.Round(Simulador.CalcularTiempo(Convert.ToDouble(dtSimTram.Rows[faseTramo][5]))
    * (i - 1), 2), Y[i - 1]);
389.                                         }
390.                                         btnOk.Enabled = true;
391.                                         //Se calcula la gráfica y se guarda la
    configuración en la Tabla de Datos 1.
392.                                         dtSimTram.Rows[faseTramo][11] = Convert.ToDo
    uble(comboDis.SelectedIndex).ToString() + "," + Convert.ToDouble(numerico3.Value)
    .ToString() + "," + Convert.ToDouble(numerico2.Value).ToString();
393.
394.                                     }
395.                                     break;
396.
397.                                 case 3://Para una distribución constante
398.                                     if (numerico3.Value > 0)
399.                                     {
400.                                         Graf.Series["Peatones"].Points.Clear();
401.                                         //Función que calcula los valores a
    representar
402.                                         int[] Y = DistribucionConstante(Convert.ToDo
    uble(numerico3.Value), 10);
403.
404.                                         for (int i = 1; i <= Y.Length; i++)
405.                                         {
406.                                             Graf.Series["Peatones"].Points.AddXY(Mat
    h.Round(Simulador.CalcularTiempo(Convert.ToDouble(dtSimTram.Rows[faseTramo][5]))
    * (i - 1), 2), Y[i - 1]);
407.                                         }
408.                                         btnOk.Enabled = true;
409.                                         //Se calcula la gráfica y se guarda la
    configuración en la Tabla de Datos 1.
410.                                         dtSimTram.Rows[faseTramo][11] = Convert.ToDo
    uble(comboDis.SelectedIndex).ToString() + "," + Convert.ToDouble(numerico3.Value)
    .ToString() + "," + Convert.ToDouble(numerico2.Value).ToString();
411.
412.                                     }
413.                                     break;
414.                                 default: break;
415.                             }
416.                             break;
417.                         default:break;
418.                     }
419.
420.
421.                 }
422.
423.                 private List<PointLatLng> Flecha(double lngI, double lngF, double la
    tI, double latF)
424.                 {
425.                     //Función para la creación de las flechas que representan Tramos
    en el Mapa
```

```
426.                 double lat, lng;
427.
428.                 List<PointLatLng> puntos = new List<PointLatLng>(); //Lista para
   almacenar todos los puntos
429.                 //Puntos Iniciales y Finales
430.                 PointLatLng PI = new PointLatLng(latI, lngI);
431.                 PointLatLng PF = new PointLatLng(latF, lngF);
432.                 //Valores característicos de las flechas
433.                 double l = Math.Sqrt(Math.Pow(lngF - lngI, 2.0) + Math.Pow(latF
   - latI, 2.0)); //Longitud
434.                 double G = -Math.Atan((latF - latI) / (lngF - lngI)); //Gamma
   (Orientación de la flecha)
435.                 double A = 0.001; //Alfa
436.                 double VarX, VarY; //Variaciones
437.                 if (lngI>lngF) G += Math.PI; //S ila orientación está entre 90 y
   270 grados, se suma valor PI
438.
439.
440.                 //Se calculan todos los puntos mediante los valores
   característicos y la rotación
441.                 VarX = 0;
442.                 VarY = l * Math.Tan(A);
443.                 lat = -VarX * Math.Sin(G) + VarY * Math.Cos(G); //Rotación en
   eje Z
444.                 lng = VarX * Math.Cos(G) + VarY * Math.Sin(G);  //Rotanción en
   eje Z
445.                 PointLatLng P1 = new PointLatLng(latI + lat, lngI + lng);
446.
447.                 VarX = l *0.8;
448.                 VarY = 0;
449.                 lat = -VarX * Math.Sin(G) + VarY * Math.Cos(G);//Rotación en eje
   Z
450.                 lng = VarX * Math.Cos(G) + VarY * Math.Sin(G);//Rotación en eje
   Z
451.                 PointLatLng P2 = new PointLatLng(P1.Lat + lat, P1.Lng + lng);
452.
453.                 VarX = 0;
454.                 VarY = l * Math.Tan(A*10);
455.                 lat = -VarX * Math.Sin(G) + VarY * Math.Cos(G);  //Rotación en
   eje Z
456.                 lng = VarX * Math.Cos(G) + VarY * Math.Sin(G);  //Rotación en
   eje Z
457.                 PointLatLng P3 = new PointLatLng(P2.Lat + lat, P2.Lng + lng);
458.
459.                 VarX = 0;
460.                 VarY = -l * Math.Tan(A);
461.                 lat = -VarX * Math.Sin(G) + VarY * Math.Cos(G);  //Rotación en
   eje Z
462.                 lng = VarX * Math.Cos(G) + VarY * Math.Sin(G);  //Rotación en
   eje Z
463.                 PointLatLng P6 = new PointLatLng(latI + lat, lngI + lng);
464.
465.                 VarX = l * 0.8;
```

```
466.                    VarY = 0;
467.                    lat = -VarX * Math.Sin(G) + VarY * Math.Cos(G);  //Rotación en
    eje Z
468.                    lng = VarX * Math.Cos(G) + VarY * Math.Sin(G);  //Rotación en
    eje Z
469.                    PointLatLng P5 = new PointLatLng(P6.Lat + lat, P6.Lng + lng);
470.
471.                    VarX = 0;
472.                    VarY = -l * Math.Tan(A*10);
473.                    lat = -VarX * Math.Sin(G) + VarY * Math.Cos(G);  //Rotación en
    eje Z
474.                    lng = VarX * Math.Cos(G) + VarY * Math.Sin(G);  //Rotación en
    eje Z
475.                    PointLatLng P4 = new PointLatLng(P5.Lat + lat, P5.Lng + lng);
476.
477.                    //Se añaden los puntos a la lista
478.                    puntos.Add(PI);
479.                    puntos.Add(P1);
480.                    puntos.Add(P2);
481.                    puntos.Add(P3);
482.                    puntos.Add(PF);
483.                    puntos.Add(P4);
484.                    puntos.Add(P5);
485.                    puntos.Add(P6);
486.
487.                    return puntos; //La función devuelve los puntos que componen la
    flecha
488.
489.                }
490.
491.                private void timer1_Tick(object sender, EventArgs e)
492.                {//Función de que se cumple en cada ciclo de timer1
493.                    //Con esta función se desea provocar una animación para aumentar
    el tamaño de la ventana
494.                    switch (expandir)
495.                    { //En función de expandir....
496.                        case 1: //Si vale 1, el formulario crece hasta 450x180 px, y
    el panel izquierdo hasta 220x225 px
497.
498.                        if (this.Size.Width < 440)
499.                        {
500.                                timer1.Interval = 10;
501.                                this.Size = new Size(this.Size.Width + (450 - this.S
    ize.Width)/10, this.Size.Height + (180 - this.Size.Height) / 10);
502.                                panel2.Size = new Size(panel2.Size.Width + (220 - pa
    nel2.Size.Width) / 10, panel2.Size.Height + (225 - panel2.Size.Height) / 10);
503.                        }
504.                        else
505.                        {
506.                            this.Size = new Size(450, 180);
507.                            panel2.Size = new Size(220, 225);
508.                            expandir = 0;
509.                            timer1.Interval = 50;
```

```
510.                     }
511.                     break;
512.                     case 2://Si vale 2, el formulario crece hasta 1150x700 px, y
    el panel izquierdo hasta 220x644 px
513.
514.                         if (this.Size.Width < 1130)
515.                         {
516.                             this.Size = new Size(this.Size.Width + (1140 - this.
    Size.Width) / 10, this.Size.Height + (700 - this.Size.Height) / 10);
517.                             panel2.Size = new Size(panel2.Size.Width + (220 - pa
    nel2.Size.Width) / 10, panel2.Size.Height + (644 - panel2.Size.Height) / 10);
518.                             timer1.Interval = 10;
519.                         }
520.                         else
521.                         {
522.                             this.Size = new Size(1150, 700);
523.                             panel2.Size = new Size(220, 644);
524.                             expandir = 0;
525.                             timer1.Enabled = false;
526.                         }
527.                         break;
528.                     default: break;
529.                 }
530.             }
531.
532.
533.             private void btnNewSim_Click(object sender, EventArgs e)
534.             {//Botón nueva simulación
535.                 //Reiniciar aplicación si se había empezado un proyecto
    anteriormente
536.                 if (fase > 0.9) Application.Restart();
537.                 //Actualización elementos
538.                 LNewSim.Visible = true;
539.                 textBox1.Visible = true;
540.                 btnOk.Visible = true;
541.                 expandir = 1;
542.                 fase = 1.0;
543.             }
544.
545.             private void btnOk_Click(object sender, EventArgs e)
546.             {
547.                 //Función que define la acción al pulsar el botón Siguiente
548.                 switch (fase)
549.                 {
550.                     case 1.0:
551.                         //Acciones para Fase = 1.0 (una vez ya se ha escrito el
    Nombre del Proyecto)
552.
553.
554.                         expandir = 2;//Se expande la ventana hasta su tamaño
    máximo
555.                         //Se activan las etiquetas y los distintos elementos de
    la interfaz que correspondan
```

```
556.                        //Se desactivan todos los elementos que no correspondan
557.                        LInstruccion1.Visible = true;
558.                        LInstruccion2.Visible = true;
559.                        LCuenta.Visible = true;
560.                        NombreProyecto = textBox1.Text.ToString(); //Se guarda
    el nombre del proyecto
561.                        LNombreProyecto.Text = LInstruccion1.Text = "Proyecto:
    "+ NombreProyecto;
562.                        LNombreProyecto.Visible = true;
563.                        textBox1.Visible = false;
564.                        LNewSim.Visible = false;
565.                        gMapControl1.Visible = true;
566.                        txtdescripcion.Visible = true;
567.                        txtlatitud.Visible = true;
568.                        txtlongitud.Visible = true;
569.                        btnAgregar.Visible = true;
570.                        label6.Visible = true;
571.                        numerico.Visible = true;
572.                        label1.Visible = true;
573.                        label2.Visible = true;
574.                        label3.Visible = true;
575.                        label11.Visible = true;
576.
577.                        fase = 1.1; //Avance a la siguiente fase
578.                        btnCambio.Visible = true;
579.                        btnOk.Enabled = false;
580.                        //Configuración del Visualizador de Bases de Datos
581.                        dataGridView1.DataSource = dtSimTram;
582.                        dataGridView1.Columns["Distancia"].Visible = false;
583.                        dataGridView1.Columns["Vehiculos"].Visible = false;
584.                        dataGridView1.Columns["VACIO2"].Visible = false;
585.                        dataGridView1.Columns["PeatonesF"].Visible = false;
586.                        LInstruccion1.Text = "Disponga los puntos iniciales y
    finales de los Tramos que desee.";
587.                        //label5.Visible = true;
588.                        label6.Visible = true;
589.                        //tipoIn.Visible = true;
590.                        tipoOut.Visible = true; //Visibilidad de lista de
    semáforos disponibles
591.                        btnOk.Enabled = false;
592.                        numerico.Value = 1;
593.                        numerico.Minimum = 1;//Valor Mínimo igual 1 debido a que
    no puede haber 0 carriles o menos
594.                       break;
595.
596.                    case 1.1:
597.                        //Acciones para Fase = 1.1 (Importancias)
598.
599.
600.                        fase = 1.2;
601.                        dataGridView1.Columns["Distancia"].Visible = true;
602.                        dataGridView1.Columns["Vehiculos"].Visible = true;
603.                        dataGridView1.Columns["VACIO2"].Visible = true;
```

```
604.                            dataGridView1.Columns["PeatonesF"].Visible = true;
605.                            Importancias = new double[dtSimTram.Rows.Count];
606.                            gMapControl1.Overlays.Clear();
607.                            gMapControl1.Visible = false; //Se esconde el mapa
608.                            btnCambio.Visible = false;
609.                            dataGridView1.Visible = false; //Se esconde el
    Visualizador de BD
610.                            btnOk.Enabled = false;
611.                            btnAgregar.Location = new Point(listaSalidas.Location.X
    - 100, listaSalidas.Location.Y);
612.                            tipoOut.Visible = false;
613.                            LInstruccion1.Text = "Definición de Importancias.";
614.                            LCuenta.Visible = false;
615.                            LInstruccion2.Text = string.Format("Relacione la entrada
    de {0} con las siguientes
    salidas:",dataGridView1.Rows[0].Cells[0].Value.ToString());
616.                            label1.Visible = false;
617.                            txtdescripcion.Visible = false;
618.                            label2.Visible = false;
619.                            txtlatitud.Visible = false;
620.                            label3.Visible = false;
621.                            txtlongitud.Visible = false;
622.                            label9.Visible = true;
623.                            label9.Location = new Point(477, 320);
624.                            numerico.Visible = true;
625.                            listaSalidas.Visible = true;
626.                            listaSalidas.Location = new Point(477, 355);

627.                            label6.Text = "Valor de Importancia(%)";
628.                            label6.Location = new Point(label6.Location.X,
    label9.Location.Y);
629.                            numerico.Minimum = 0;
630.                            numerico.Maximum = 100;
631.                            numerico.Value = 0;
632.                            numerico.Location= new Point(numerico.Location.X,
    listaSalidas.Location.Y);
633.                            label11.Visible = false;
634.
635.                            faseTramo = 0;
636.                            limpiarArray(); //Vecotor Importancias a 0
637.                            //Se llena la lista Saliddas con todos Tramos
638.                            for (int filas = 0; filas < dataGridView1.Rows.Count; fi
    las++)
639.                            {
640.                                Salidas.Add(Convert.ToString(dataGridView1.Rows[fila
    s].Cells[0].Value));
641.                                listaSalidas.Items.Add(Salidas[filas].ToString());
642.                            }
643.                            //Se elimina de la lista el Tramo que se pregunta
644.                            listaSalidas.Items.RemoveAt(faseTramo);
645.
646.
647.
```

```
                        break;


                    case 1.3: // Fase de Flujo de Vehículos
                        //Nueva posición botón agregar
                        btnAgregar.Location = new Point(btnOk.Location.X + 100,
   btnOk.Location.Y+4);
                        btnAgregar.Visible = true;
                        btnAgregar.Enabled = true;
                        if (faseTramo < 0)
                        {
                            //Si todavía no se ha empezado, disponer los
   elementos en pantalla
                            // y preguntar el flujo de vehículos de la primera
   entrada principal
                            faseTramo = 0;
                            label9.Visible = false;
                            listaSalidas.Visible = false;
                            dataGridView1.DataSource = dtSimTram;
                            dataGridView1.SendToBack();
                            gMapControl1.SendToBack();
                            comboDis.Visible = true;
                            ListLog.Controls.Add(this.Asc);//Radiobutton para
   orden ascendente
                            ListLog.Controls.Add(this.Desc);//Radiobutton para
   orden descendente
                            //Situación de elementos
                            comboDis.BringToFront();
                            numerico3.BringToFront();
                            numerico2.BringToFront();
                            label13.BringToFront();
                            label12.BringToFront();
                            btnAgregar.BringToFront();
                            label6.Visible = false;
                            numerico.Visible = false;
                            label13.Text = "Media";
                            label12.Text = "Desviación";
                            btnOk.Enabled = false;
                            //Busqueda de la primera entrada principal
                            while (faseTramo < dtSimTram.Rows.Count && Convert.T
   oDouble(dtSimRel.Rows[faseTramo][2])!=0.0) { faseTramo++; }
                            LInstruccion1.Text = "Determinación del flujo de
   Vehículos para las Entradas Principales.";
                            try { LInstruccion2.Text = String.Format("Flujo de
   vehículos para la entrada: {0}.", dtSimTram.Rows[faseTramo][0].ToString()); }
                            catch { MessageBox.Show("Importancias Erróneas. No
   hay ninguna entrada principal.");
                                Application.Restart();
                                    }

                        }
                        else if (faseTramo>= 0 && faseTramo < dtSimTram.Rows.Cou
   nt)
```

```
691.                         {
692.                             //Si ya se ha empezado la fase, pues se prosigue
693.                             faseTramo++;//Incremento del Tramo que se pregunta
    para poder proseguir
694.                             //Se busca la próxima entrada principal
695.                             while (faseTramo < dtSimTram.Rows.Count && Convert.T
    oDouble(dtSimRel.Rows[faseTramo][2]) != 0.0) { faseTramo++; }
696.                             if (faseTramo < dtSimTram.Rows.Count) //Si el Tramo
    existe, preguntar por su flujo
697.                             {
698.                                 btnOk.Enabled = false;
699.                                 comboDis.SelectedIndex = 0;
700.                                 LInstruccion2.Text = String.Format("Flujo de
    vehículos para la entrada: {0}.",
    dtSimTram.Rows[faseTramo][0].ToString()/*dtSimTram.Rows[faseTramo][0].ToString()*
    /);
701.                                 numerico3.Value = 0;
702.                                 numerico2.Value = 0;
703.                                 Graf.Series["Vehículos"].Points.Clear();
704.                                 Asc.Checked = false;
705.                                 Desc.Checked = false;
706.                             }
707.                             else //Si no existe, hay que avanzar a la próxima
    fase
708.                             {
709.                                 faseTramo = 0; //Tramo de estudio =0
710.                                 fase = 1.4; //Avance de fase
711.                                 label5.Visible = false;
712.                                 btnOk.Enabled = false;
713.                                 comboDis.SelectedIndex = 0;
714.                                 //La etiqueta informa que se piden los peatones
715.                                 LInstruccion1.Text = "Determinación del flujo de
    Peatones para las Entradas Principales.";
716.                                 LInstruccion2.Text = String.Format("Flujo de
    peatones para: {0}.", dtSimTram.Rows[faseTramo][0].ToString());
717.                                 numerico3.Value = 0;
718.                                 numerico2.Value = 0;
719.                                 //Reseteo de la gráfica
720.                                 Graf.Series["Vehículos"].Points.Clear();
721.                                 Graf.Series.Clear();
722.                                 //Creación de la serie Peatones
723.                                 Graf.Series.Add("Peatones");
724.                                 Graf.Series["Peatones"].ChartType = SeriesChartT
    ype.Line; //Tipo de gráfica
725.                                 //Desactivación de los radiobuttons
726.                                 Asc.Checked = false;
727.                                 Desc.Checked = false;
728.                             }
729.                         }
730.                             break;
731.
732.                     case 1.4: //Fase de Peatones
733.
```

```
734.                         faseTramo++; //Se incrementa el Tramo de estudio
735.                         if (faseTramo >= 0 && faseTramo < dtSimTram.Rows.Count)
736.                         {
737.                             //Si el tramo existe, se pregunta por su flujo de
    peatones
738.                             LInstruccion1.Text = "Determinación del flujo de
    Peatones para todos los Tramos.";
739.                             LInstruccion2.Text = String.Format("Densidad de
    peatones para el Tramo: {0}", dtSimTram.Rows[faseTramo][0].ToString());
740.                             //Reseteo de los elementos
741.                             btnOk.Enabled = false;
742.                             comboDis.SelectedIndex = 0;
743.                             numerico3.Value = 0;
744.                             numerico2.Value = 0;
745.                             Graf.Series["Peatones"].Points.Clear();
746.                             Asc.Checked = false;
747.                             Desc.Checked = false;
748.                         }
749.                         else if (faseTramo >= dtSimTram.Rows.Count)
750.                         {
751.                             //Si no existe, se procede a avanzar en la
    aplicación
752.                             faseTramo = -1;
753.                             fase = 1.5;
754.                             //Se inicia el guardado de la configuración
755.                             if (folderBrowserDialog1.ShowDialog() == DialogResul
    t.OK)
756.                             {
757.                                 //Si la selección de la carpeta es correcta, se
    guarda la dirección
758.
759.                                 dir = @folderBrowserDialog1.SelectedPath+"\\Conf
    ig_"+NombreProyecto+".txt";
760.                                 CrearConfig(); //Función que crea el Archivo de
    Texto con la configuración
761.                                 //Redisposición de los distintos elementos de la
    Interfaz para preparar la simulación
762.                                 btnOk.Enabled = false;
763.                                 btnOk.Visible = false;
764.                                 btnAgregar.Visible = false;

765.                                 bPlay.Visible = true;
766.                                 bPlay.Enabled = true;
767.                                 bDetener.Enabled = true;
768.                                 bDetener.Visible = true;
769.                                 bFinalizar.Enabled = true;
770.                                 bFinalizar.Visible = true;
771.                                 gMapControl1.Overlays.Clear();
772.                                 gMapControl1.Visible = true;
773.                                 dataGridView1.Visible = false;
774.                                 gMapControl1.Enabled = true;
775.                                 IniciarSimulacion();
776.                                 iteracion = 0;
```

```
777.                                    trackBar1.Value = 1;
778.                                    trackBar1.Visible = true;
779.                                    label13.Location = new Point(trackBar1.Location.
    X + 5, trackBar1.Location.Y - 25);
780.                                    label13.Visible = true;
781.                                    label13.BringToFront();
782.                                    label13.Text = String.Format("Velocidad x{0}",
    trackBar1.Value);
783.                                    label5.Visible = false;
784.                                    LCuenta.Visible = false;
785.                                    LInstruccion1.Text = "Simulación.";
786.                                    LInstruccion2.Visible = false;
787.                                }
788.
789.                            }
790.
791.
792.                            break;
793.                        default:break;
794.
795.                    }
796.
797.
798.
799.                }
800.
801.                private void btnCambio_Click(object sender, EventArgs e)
802.                {
803.                    //Función para el botón de Cambiar Vista (Visualizador<->Mapa)
804.                    if(gMapControl1.Visible == true)
805.                    {
806.                     //Si el Mapa es visible, este desaparece por el Visualizador de
    Bases de Datos
807.                        gMapControl1.Visible = false;
808.                        dataGridView1.Visible = true;
809.                    }
810.                    else
811.                    {
812.                        //Si el Visualizador es visible, este desaparece por el Mapa
813.                        gMapControl1.Visible = true;
814.                        dataGridView1.Visible = false;
815.                    }
816.                }
817.
818.                private void textBox1_TextChanged_1(object sender, EventArgs e)
819.                {
820.                    //Al escribir algo en el Cuadro de texto, se habilita el botón
    Siguiente
821.                    //Etapa 2
822.                    btnOk.Enabled = true;
823.                }
824.
825.                private void numerico_ValueChanged(object sender, EventArgs e)
```

```
826.             {
827.                 //Cuando varía el valor del seleccionador numérico, se lleva a
   cabo el guardado
828.                 //de las Importancias
829.                 if(fase == 1.2) {
830.                         if (Agregar) {
831.                             Agregar = false;
832.                         }
833.                         else if(listaSalidas.SelectedIndex >= faseTramo && lista
   Salidas.SelectedIndex>=0)
834.                         {
835.                             //Condicional para evitar solapes. No atribuir
   importancias en la zona de estudio
836.                             Importancias[listaSalidas.SelectedIndex+1] = Convert
   .ToDouble(numerico.Value);
837.                         }
838.                         else if(listaSalidas.SelectedIndex >= 0)
839.                         {
840.                             Importancias[listaSalidas.SelectedIndex] = Convert.T
   oDouble(numerico.Value);
841.                         }
842.
843.                 }
844.             }
845.
846.             private void txtlatitud_KeyPress(object sender, KeyPressEventArgs e)
847.             {
848.                 //Función para que el usuario escriba correctamente la velocidad
   media
849.                 char C = e.KeyChar;
850.                 if (txtVelMed.Text.IndexOf('.') != -1 && C == 46) //Para no
   poder escribir 2 veces el punto
851.                 {
852.                     e.Handled = true;
853.                     return;
854.                 }
855.                 if (C != 8 && C != 46 && !Char.IsDigit(C)) //No aceptar
   caracteres no numericos (excepto signos)
856.                 {
857.                     e.Handled = true;
858.                 }
859.             }
860.
861.             private void txtlongitud_KeyPress(object sender, KeyPressEventArgs
   e)
862.             {
863.                 //Función para que el usuario escriba correctamente la velocidad
   media
864.                 char C = e.KeyChar;
865.                 if (txtVelMed.Text.IndexOf('.') != -1 && C == 46) //Para no
   poder escribir 2 veces el punto
866.                 {
867.                     e.Handled = true;
```

```
868.                     return;
869.                 }
870.                 if (C != 8 && C != 46 && !Char.IsDigit(C)) //No aceptar
    caracteres no numéricos (excepto signos)
871.                 {
872.                     e.Handled = true;
873.                 }
874.             }
875.
876.             private void comboDis_SelectedIndexChanged(object sender, EventArgs
    e)
877.             {
878.
879.                 label5.BringToFront();
880.                 //Función activada al variar el valor seleccionado del Cuadro
    Combinado
881.                 switch (comboDis.SelectedIndex)
882.                 {
883.                     case 1: //Se ha seleccionado la distribución normal
884.                 //Disposición de los elementos de Interfaz necesarios para
    observar esta distribución
885.                         Graf.Visible = true;
886.                         btnAgregar.Visible = true;
887.                         numerico3.Visible = true;
888.                         label13.Visible = true;
889.                         label12.Visible = true;
890.                         numerico2.Visible = true;
891.                         btnAgregar.Enabled = true;
892.                         ListLog.Visible = false;
893.                         Asc.Visible = false;
894.                         Desc.Visible = false;
895.                         label5.Visible = true;
896.                         break;
897.                     case 2: //Se ha seleccionado la distribución en Rampa
898.                 //Disposición de los elementos de Interfaz necesarios para
    observar esta distribución
899.                         numerico2.Visible = false;
900.                         label12.Visible = false;
901.                         label13.Visible = true;
902.                         label13.Text = "Media";
903.                         numerico3.Visible = true;
904.                         btnAgregar.Enabled = true;
905.                         ListLog.Visible = true;
906.                         Asc.Visible = true;
907.                         Desc.Visible = true;
908.                         label5.Visible = true;
909.                         break;
910.                     case 3://Se ha seleccionado Distribución Constante
911.                 //Disposición de los elementos de Interfaz necesarios para
    observar esta distribución
912.                         numerico2.Visible = false;
913.                         label12.Visible = false;
914.                         label13.Visible = true;
```

```
915.                            label13.Text = "Media";
916.                            numerico3.Visible = true;
917.                            btnAgregar.Enabled = true;
918.                            ListLog.Visible = false;
919.                            Asc.Visible = false;
920.                            Desc.Visible = false;
921.                            label5.Visible = true;
922.                            break;
923.                        default:
924.                            break;
925.                    }
926.                }
927.
928.                private void numerico_KeyPress(object sender, KeyPressEventArgs e)
929.                {
930.                    //Si se cambia el valor de importancias por teclado, este valor
    se guarda
931.                    if (fase == 1.2)
932.                    {
933.                        if (Agregar)
934.                        {
935.                            Agregar = false; //No hacer caso si se trata de un
    cambio del programa.
936.                        }
937.                        else if (listaSalidas.SelectedIndex >= faseTramo && listaSal
    idas.SelectedIndex >= 0)
938.                        {
939.                            //Condicional para evitar solapes. No atribuir
    importancias en la zona de estudio
940.                            Importancias[listaSalidas.SelectedIndex + 1] = Convert.T
    oDouble(numerico.Value);
941.                        }
942.                        else if (listaSalidas.SelectedIndex >= 0)
943.                        {
944.                            Importancias[listaSalidas.SelectedIndex] = Convert.ToDou
    ble(numerico.Value);
945.                        }
946.
947.                    }
948.                }
949.
950.                private void IniciarSimulacion()
951.                {
952.                    //Función con la que se lleva a cabo el inicio de la Simulación
953.                    dataGridView1.DataSource = dtSimTram;
954.                    //Nombres de los Tramos
955.                    string[] Nombres = new string[dataGridView1.RowCount];
956.                    //Vector entero para indicar si son entradas principales (1->Si,
    0->No)
957.                    int[] EntradasP = new int[dtSimRel.Rows.Count];
958.                    //Vector double donde se guardan las distancias
959.                    double[] Distancias = new double[dataGridView1.RowCount];
960.                    //Valores enteros a tener en cuenta
```

```
961.                 int VehiculosEnTramo, PeatonesEnTramo, CiclosRestantes, Tipo;
962.                 //Variable que define el Estado del Tramo (true->Activo, false-
   >Parado)
963.                 bool Estado;
964.     //Se guardan en los vectores, los nombres y las distancias almacenados en la
   Tabla de Datos 1
965.                 for (int filas = 0; filas < dataGridView1.RowCount; filas++)
966.                 {
967.                     Nombres[filas] = Convert.ToString(dataGridView1.Rows[filas].
   Cells[0].Value);
968.                     //Distancia en metros
969.                     Distancias[filas] = Convert.ToDouble(dataGridView1.Rows[fila
   s].Cells[8].Value)*1000.0;
970.                 }
971.                 dataGridView1.DataSource = dtSimRel;
972.                 //Se define si cada Tramo es una Entrada Principal en función de
   la Suma de Importancias
973.                 for (int filas = 0; filas < dtSimRel.Rows.Count; filas++)
974.                 {
975.                     if (Convert.ToDouble(dtSimRel.Rows[filas][2]) > 0.0)
976.                         EntradasP[filas] = 0;
977.                     else EntradasP[filas] = 1;
978.                 }
979.                 //Se ocultan los elementos de Interfaz de la Etapa Anterior
980.                 comboDis.Visible = false;
981.                 Graf.Visible = false;
982.                 label13.Visible = false; //Label de Media
983.                 numerico3.Visible = false;
984.                 ListLog.Visible = false;
985.                 Desc.Visible = false;
986.                 Asc.Visible = false;
987.                 numerico2.Visible = false;
988.                 label12.Visible = false;
989.
990.                 double LatI, LngI, LatF, LngF; //Variables de Coordenadas Punto
   Inicial y Final
991.                 //Se obtiene la Tabla de Datos 3, a partir de la Instancia
   Simulador, de la Clase Algoritmo
992.                 Sim = Simulador.Operaciones(dtSimTram, dtSimRel, Sim,
   dataGridView1, -1, Convert.ToDouble(Temporizador.Interval),EntradasP,
   Distancias);
993.
994.                 LCuenta.Text = EntradasP.Length.ToString();
995.                 LCuenta.Visible = true;
996.                 //Se inicia un bucle para representar cada uno de los Tramos en
   el Mapa
997.                 for (int filas = 0; filas < dtSimTram.Rows.Count; filas++)
998.                 {
999.                     //Obtención de cada uno de los valores a partir de la Tabla
   de Datos 1
1000.                    LatI = Convert.ToDouble(dtSimTram.Rows[filas][1]);
1001.                    LngI = Convert.ToDouble(dtSimTram.Rows[filas][2]);
1002.                    LatF = Convert.ToDouble(dtSimTram.Rows[filas][3]);
```

```
1003.                     LngF = Convert.ToDouble(dtSimTram.Rows[filas][4]);
1004.                     Tipo = Convert.ToInt16(dtSimTram.Rows[filas][6]);
1005.                     Estado = Convert.ToBoolean(Sim.Rows[filas][0]);
1006.                     CiclosRestantes = Convert.ToInt32(Sim.Rows[filas][2]);
1007.                     VehiculosEnTramo = Convert.ToInt32(Sim.Rows[filas][4]);
1008.                     PeatonesEnTramo = Convert.ToInt32(Sim.Rows[filas][5]);
1009.
1010.
1011.                     //Creación del polígono (flecha)
1012.                     GMapOverlay Poligono = new GMapOverlay(Nombres[filas]);
1013.                     //Obtención de los puntos que componen la flecha
1014.                     List<PointLatLng> puntos = Flecha(LngI,LngF,LatI,LatF);
1015.                     GMapPolygon poligonoPuntos = new GMapPolygon(puntos,
   Nombres[filas]);
1016.                     Poligono.Polygons.Add(poligonoPuntos);
1017.                     //Situación del marcador justo en mitad de la flecha
1018.                     marker = new GMarkerGoogle(new PointLatLng((LatF + LatI) * 0
   .5, (LngF + LngI) * 0.5), GMarkerGoogleType.white_small);
1019.
1020.
1021.
1022.                     //agregamos un toolip de texto a los marcadores.
1023.                     marker.ToolTipMode = MarkerTooltipMode.OnMouseOver; //Solo
   se muestra con el Ratón encima
1024.                     //Se muestra con el marcador, el nombre
1025.                     marker.ToolTipText = string.Format("{0}", Nombres[filas]);
1026.                     //marker.ToolTipMode = MarkerTooltipMode.Always;
1027.                     //marker.ToolTipText = string.Format("{0}", Nombres[filas]);
1028.                     Poligono.Markers.Add(marker);
1029.                     gMapControl1.Overlays.Add(Poligono);
1030.                 }
1031.                 //Incremento y decremento del zoom para actualizar el mapa
1032.                 gMapControl1.Zoom += 1;
1033.                 gMapControl1.Zoom -= 1;
1034.             }
1035.             private void Simular()
1036.             {
1037.                 //Función con la que se lleva a cabo el inicio de la Simulación
1038.                 dataGridView1.DataSource = dtSimTram;
1039.                 //Variable Estado del Tramo (true->Activo, False->Parado)
1040.                 //Variable Alternancia para provocar el efecto intermitente del
   ambar (negro y ámbar)
1041.                 bool Estado,Alternancia;
1042.                 //Variables de Interés, UV-> Vehículos que van a abandonar el
   Tramo en el siguinte periodo
1043.                 int VehiculosEnTramo, PeatonesEnTramo, CiclosRestantes, Tipo,UV;
1044.                 //Nombres de los Tramos
1045.                 string[] Nombres = new string[dtSimTram.Rows.Count];
1046.                 //Vector entero para indicar si son entradas principales (1->Si,
   0->No)
1047.                 int[] EntradasP = new int[dtSimRel.Rows.Count];
1048.                 //Vector Doble que almacena las distancias
1049.                 double[] Distancias = new double[dtSimTram.Rows.Count];
```

```
1050.    //Se guardan en los vectores, los nombres y las distancias almacenados en la
   Tabla de Datos 1
1051.                for (int filas = 0; filas < dataGridView1.RowCount; filas++)
1052.                {
1053.                    Nombres[filas] = Convert.ToString(dtSimTram.Rows[filas][0]);
1054.                    Distancias[filas] = Convert.ToDouble(dtSimTram.Rows[filas][8
   ]) * 1000.0;
1055.                }
1056.                //Se define si cada Tramo es una Entrada Principal en función de
   la Suma de Importancias
1057.                for (int filas = 0; filas < dtSimRel.Rows.Count; filas++)
1058.                {
1059.                    if (Convert.ToDouble(dtSimRel.Rows[filas][2]) > 0.0)
1060.                        EntradasP[filas] = 0;
1061.                    else EntradasP[filas] = 1;
1062.                }
1063.                //Se obtiene la Tabla de Datos 3, a partir de la Instancia
   Simulador, de la Clase Algoritmo
1064.                //Los valores que ocupan Sim, son los valores actualizados de
   todos los Tramos, en cada iteración.
1065.                Sim = Simulador.Operaciones(dtSimTram, dtSimRel, Sim,
   dataGridView1, iteracion, Convert.ToDouble(Temporizador.Interval), EntradasP,
   Distancias);
1066.                dataGridView1.DataSource = Sim;
1067.                double LatI, LngI, LatF, LngF;//Variables de coordenadas Puntos
   Iniciales y Finales
1068.
1069.
1070.                //Se inicia un bucle para actualizar cada uno de los Tramos en
   el Mapa
1071.                for (int filas = 0; filas < dataGridView1.RowCount; filas++)
1072.                {
1073.                    LatI = Convert.ToDouble(dtSimTram.Rows[filas][1]);
1074.                    LngI = Convert.ToDouble(dtSimTram.Rows[filas][2]);
1075.                    LatF = Convert.ToDouble(dtSimTram.Rows[filas][3]);
1076.                    LngF = Convert.ToDouble(dtSimTram.Rows[filas][4]);
1077.                    Tipo = Convert.ToInt16(dtSimTram.Rows[filas][6]);
1078.                    Estado = Convert.ToBoolean(Sim.Rows[filas][0]);
1079.                    CiclosRestantes = Convert.ToInt32(Sim.Rows[filas][2]);
1080.                    VehiculosEnTramo = Convert.ToInt32(Sim.Rows[filas][4]);
1081.                    PeatonesEnTramo = Convert.ToInt32(Sim.Rows[filas][5]);
1082.                    UV = Convert.ToInt32(Sim.Rows[filas][8]);
1083.                    Alternancia = Convert.ToBoolean(Sim.Rows[filas][11]);
1084.
1085.                    //Representacíón de las flechas y marcadores.
1086.                    GMapOverlay Poligono = new GMapOverlay(Nombres[filas]);
1087.                    //Obtención de los puntos de las flechas.
1088.                    List<PointLatLng> puntos = Flecha(LngI, LngF, LatI, LatF);
1089.                    GMapPolygon poligonoPuntos = new GMapPolygon(puntos,
   Nombres[filas]);
1090.                    Poligono.Polygons.Add(poligonoPuntos);
1091.                    //Se llama la función Marcadores, que devuelve un marker con
   el color que debe tener
```

```
1092.                     //Dado que el color varía si es un semáforo de rojo, ámbar y
   verde o de otro tipo
1093.                     //además de los ciclos restantes
1094.                     marker = Marcadores(Estado, Tipo, LatI, LngI, LatF, LngF,
   CiclosRestantes,Alternancia);
1095.                     gMapControl1.UpdateMarkerLocalPosition(marker);
1096.
1097.                     //agregamos un toolip de texto a los marcadores.
1098.                     //Los marcadores solo se muestran con el ratón encima
1099.                     marker.ToolTipMode = MarkerTooltipMode.OnMouseOver;
1100.                     //Muestran los valores de interés: Vehículos, Peatones,
   Ciclos Restantes...
1101.                     marker.ToolTipText = string.Format("{0}_F\nVehiculos:{1}\nPe
   atones:{2}\nCiclosR:{3}\nIteracion:{4}\nUV:{5}", Nombres[filas],
   VehiculosEnTramo, PeatonesEnTramo, CiclosRestantes, iteracion, UV);
1102.                     Poligono.Markers.Add(marker);
1103.                     gMapControl1.Overlays.Add(Poligono);
1104.
1105.                 }
1106.             }
1107.
1108.
1109.             private void Desc_CheckedChanged(object sender, EventArgs e)
1110.             {
1111.                 //Si el RadioButton de Rampa Descendente se pulsa, se desactiva
   el de Ascendente
1112.                 if (Desc.Checked) Asc.Checked = false;
1113.             }
1114.
1115.             private void Asc_CheckedChanged(object sender, EventArgs e)
1116.             {
1117.                 //Si el RadioButton de Rampa Ascendente se pulsa, se desactiva
   el de Descendente
1118.                 if (Asc.Checked) Desc.Checked = false;
1119.             }
1120.
1121.             private void bPlay_Click(object sender, EventArgs e)
1122.             {
1123.                 //Función Botón Play
1124.
1125.                 //Preparación vector capacidades para emplearlo posteriormente
1126.                 Capacidades = new string[dtSimTram.Rows.Count];
1127.                 //El forecolor se cambiar, para que solo aparezca el botón Play
   como pulsado
1128.                 bPlay.ForeColor = Color.FromArgb(0, 80, 200);
1129.                 bDetener.ForeColor = Color.FromArgb(49, 66, 82);
1130.                 bFinalizar.ForeColor = Color.FromArgb(49, 66, 82);
1131.                 Capacidades[0] = Convert.ToInt32(Convert.ToDouble(dtSimTram.Rows
   [0][8]) * 1000.0 * Convert.ToInt32(dtSimTram.Rows[0][7]) / 5).ToString();
1132.                 //Se llena el vector capacidades, suponiendo que cada vehículo
   ocupa 5metros
1133.                 //y teniendo en cuenta el n° de carriles y la distancia (en
   metros)
```

```
1134.                 for (int filas= 1;filas < dtSimTram.Rows.Count; filas++)
1135.                 {
1136.                    Capacidades[filas] = Convert.ToInt32(Convert.ToDouble(dtSimTr
      am.Rows[filas][8]) * 1000.0 * Convert.ToInt32(dtSimTram.Rows[filas][7]) / 5).ToSt
      ring();
1137.                 }
1138.                 //Se activa el Temporizador
1139.                 Temporizador.Enabled = true;
1140.                 Temporizador.Start();
1141.
1142.             }
1143.
1144.             private void bDetener_Click(object sender, EventArgs e)
1145.             {//Función del botón Detener (Pause)
1146.                 Temporizador.Stop();//Se para el temporizador
1147.                 //Se dispone que el botón Detener aparezca como pulsado
1148.                 bPlay.ForeColor = Color.FromArgb(49, 66, 82);
1149.                 bDetener.ForeColor = Color.FromArgb(0, 80, 200);
1150.                 bFinalizar.ForeColor = Color.FromArgb(49, 66, 82);
1151.             }
1152.
1153.             private void bFinalizar_Click(object sender, EventArgs e)
1154.             {//Función botón Finalizar (Terminar)
1155.
1156.                 //Aparece el botón Finalizar como pulsado
1157.                 bPlay.ForeColor = Color.FromArgb(49, 66, 82);
1158.                 bDetener.ForeColor = Color.FromArgb(49, 66, 82);
1159.                 bFinalizar.ForeColor = Color.FromArgb(0, 80, 200);
1160.
1161.                 Temporizador.Stop();//Se para el temporizador
1162.                 Temporizador.Enabled = false; //Se desactiva el temporizador
1163.                 GuardarResultados(); //Función para exportar Resultados
1164.                 Sim.Clear(); //Se limpia la Tabla de Datos 3(Que contiene datos
      de Simulación)
1165.                 Resultados.Clear();//Se limpia la Tabla de Datos 4
1166.                 trackBar1.Value = 1; //El control deslizante vuelve al reposo
1167.                 gMapControl1.Overlays.Clear();//Reseteo del Mapa
1168.                 gMapControl1.Visible = true;
1169.                 dataGridView1.Visible = false;
1170.                 gMapControl1.Enabled = true;
1171.                 IniciarSimulacion();//Se vuelve a inicializar la simulación
1172.                                     //a la espera de que el usuario la pueda
      comenzar
1173.                 label13.Location = new Point(trackBar1.Location.X + 5,
      trackBar1.Location.Y - 25);
1174.                 label13.Visible = true;
1175.                 label13.BringToFront();
1176.                 label13.Text = String.Format("Velocidad x{0}", trackBar1.Value);
1177.                 iteracion = 0;
1178.                 LCuenta.Visible = false;
1179.
1180.             }
1181.
```

```
1182.            private void trackBar1_ValueChanged(object sender, EventArgs e)
1183.            {
1184.                //Si varía el valor del control deslizante, su valor afecta
   inversamente
1185.                //a la duración del periodo del temporizador
1186.                Temporizador.Interval = 1000 / trackBar1.Value;
1187.                label13.Text = String.Format("Velocidad x{0}", trackBar1.Value);
1188.            }
1189.
1190.            private void txtVelMed_KeyPress(object sender, KeyPressEventArgs e)
1191.            {
1192.                //Función para que el usuario escriba correctamente la velocidad
   media
1193.                char C = e.KeyChar;
1194.                if(txtVelMed.Text.IndexOf('.') !=-1 && C == 46) //Para no poder
   escribir 2 veces el punto
1195.                {
1196.                    e.Handled = true;
1197.                    return;
1198.                }
1199.                if(C != 8 && C != 46 && !Char.IsDigit(C)) //No aceptar
   caracteres no numericos (excepto signos)
1200.                {
1201.                    e.Handled = true;
1202.                }
1203.            }
1204.
1205.            private void listaSalidas_SelectedIndexChanged(object sender,
   EventArgs e)
1206.            {//Cuando se cambia de Tramo, se guardan las importancias.
1207.                if(fase == 1.2)
1208.                {
1209.
1210.                    if (listaSalidas.SelectedIndex >= faseTramo && listaSalidas.
   SelectedIndex >= 0)
1211.                    {
1212.                        MessageBox.Show((listaSalidas.SelectedIndex+1).ToString(
   ) + "," + Convert.ToInt32(Importancias[listaSalidas.SelectedIndex + 1]).ToString(
   ));
1213.                        numerico.Value = Convert.ToInt32(Importancias[listaSalid
   as.SelectedIndex+1]);
1214.                        Agregar = true;
1215.
1216.                    }
1217.                    else if (listaSalidas.SelectedIndex >= 0)
1218.                    {
1219.                        MessageBox.Show(listaSalidas.SelectedIndex.ToString() +
   "," + Convert.ToInt32(Importancias[listaSalidas.SelectedIndex]).ToString());

1220.                        numerico.Value = Convert.ToInt32(Importancias[listaSalid
   as.SelectedIndex]);
1221.                        Agregar = true;
1222.
```

```
1223.                     }
1224.
1225.
1226.                 }
1227.             }
1228.
1229.             private void Temporizador_Tick(object sender, EventArgs e)
1230.             {
1231.                 //Función a realizar en cada ciclo del Temporizador
1232.
1233.                 if (iteracion >= 0)
1234.                 {
1235.                     iteracion++;
1236.                     Simular(); //Se actualizan los marcadores del Mapa
1237.
1238.                     //Se obtienen los valores de las distintas variables de
     interés
1239.                     //para guardarlas en la Tabla de Datos 4(Resultados) y así
     poder exportarlas
1240.                     //Para ello se pasan todas las variables a cadenas(string)
1241.                     if (Convert.ToBoolean(Sim.Rows[0][0])) Estados = "1";
1242.                     else Estados = "0";
1243.                     Vehiculos = Convert.ToInt32(Sim.Rows[0][4]).ToString();
1244.                     Peatones = Convert.ToInt32(Sim.Rows[0][5]).ToString();
1245.                     CiclosR = Convert.ToInt32(Sim.Rows[0][2]);
1246.                     Tiempo = Convert.ToDouble(Sim.Rows[0][10]);
1247.                     TiemposEstado = (Tiempo * CiclosR).ToString();
1248.                     for (int filas = 1; filas< Sim.Rows.Count; filas++)
1249.                     {
1250.                         //Se lleva a cabo la adhesión de una coma para ayudar a
     la lectura de los
1251.                         //datos una vez están en el Archivo de Texto
1252.                         if (Convert.ToBoolean(Sim.Rows[filas][0])) Estados += ",
     1";
1253.                         else Estados += ",0";
1254.                         CiclosR = Convert.ToInt32(Sim.Rows[filas][2]);//Ciclos
     Restantes
1255.                         //Tiempo que al que avanza el vector de cada tramo
1256.                         Tiempo = Convert.ToDouble(Sim.Rows[filas][10]);
1257.                         TiemposEstado += "," + (Tiempo * CiclosR).ToString();//T
     iempo Restante
1258.                         Vehiculos += "," +Convert.ToInt32(Sim.Rows[filas][4]).To
     String();//Vehículo en el Tramo
1259.                         Peatones += ","+Convert.ToInt32(Sim.Rows[filas][5]).ToSt
     ring(); //Peatones en el Tramo
1260.                     }
1261.                     //Todas las cadenas (string) de interés se almacenan en la
     Tabla de Datos 4 (Resultados)
1262.                     Resultados.Rows.Add(iteracion, Estados, Vehiculos, Peatones,
     Capacidades,TiemposEstado);
1263.                 }
1264.             }
1265.
```

```
1266.            private void btnLoadSim_Click(object sender, EventArgs e)
1267.            {//Función al pulsar botón Cargar Simulación
1268.
1269.                //Si la aplicación ya estaba en funcionamiento, esta se reinicia
1270.                if (fase > 0.9) Application.Restart();
1271.                //Se prepara la ventana emeregente para cargar el Archivo de
   Texto de Configuración
1272.                //Instancia que contiene el cuadro de diálogo
1273.                if (fase <= 0.9)
1274.                {
1275.                    OpenFileDialog openFileDialog1 = new OpenFileDialog();
1276.                    openFileDialog1.InitialDirectory = "c:\\"; //Directorio que
   aparece al iniciarse
1277.                    openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files
   (*.*)|*.*";//Tipo de archivo
1278.                    openFileDialog1.FilterIndex = 2; //Índice de filtro
1279.                    openFileDialog1.RestoreDirectory = true; //Restaura el
   directorio antes de cerrarse
1280.
1281.                    if (openFileDialog1.ShowDialog() == DialogResult.OK)
1282.                    {
1283.                        //Si se selecciona un archivo correcto
1284.                        try
1285.                        {
1286.                            dir = openFileDialog1.FileName; //Obtiene dirección
1287.
1288.                            LeerConfig();//Función que lee la configuración
   almacenada
1289.                            expandir = 2;//La ventana aumenta al máximo tamaño
1290.                            faseTramo = -1;//Tramo -1 para poder iniciar
   correctamente
1291.                            fase = 1.5; //Al cargar la configuración, se reanuda
   desde la etapa Simulación
1292.                            btnOk.Enabled = false;
1293.                            ///AQUI EMPIEZAS A PREPARAR EL FORMULARIO PARA
   SIMULAR
1294.                            bPlay.Visible = true;
1295.                            bPlay.Enabled = true;
1296.                            bDetener.Enabled = true;
1297.                            bDetener.Visible = true;
1298.                            bFinalizar.Enabled = true;
1299.                            bFinalizar.Visible = true;
1300.                            gMapControl1.Overlays.Clear();
1301.                            gMapControl1.Visible = true;
1302.                            dataGridView1.Visible = false;
1303.                            gMapControl1.Enabled = true;
1304.                            IniciarSimulacion();
1305.                            iteracion = 0;
1306.                            trackBar1.Value = 1;
1307.                            trackBar1.Visible = true;
1308.                            label13.Location = new Point(trackBar1.Location.X +
   5, trackBar1.Location.Y - 25);
1309.                            label13.Visible = true;
```

```
1310.                               label13.BringToFront();
1311.                               label13.Text = String.Format("Velocidad x{0}",
     trackBar1.Value);
1312.                               LCuenta.Visible = false;
1313.                               LInstruccion1.Text = "Simulación.";
1314.                               LInstruccion1.Visible = true;
1315.                               LNombreProyecto.Text = String.Format("Proyecto:
     {0}", NombreProyecto);
1316.
1317.                           }
1318.                           catch (Exception ex)
1319.                           {
1320.                               //En caso de error, mostrar el siguiente mensaje.
1321.                               MessageBox.Show("Error: Archivo no compatible
     " + ex.Message);
1322.                           }
1323.
1324.                       }
1325.                   }
1326.               }
1327.
1328.               void limpiarArray()
1329.               {//Función para establecer todos los valores del vector importancias
     a 0
1330.                   for (int i = 0; i < dtSimTram.Rows.Count; i++)
1331.                   {
1332.                      Importancias[i] = 0.0;
1333.                   }
1334.               }
1335.               int[] DistribucionNormal(double Media, double Desviacion, int tiempo
     )
1336.               {
1337.                   //Función de la Expresión de la Distribución Normal según la
     media y la desviación
1338.                   tiempo++; //Para ajustar al tiempo real 10 -> tiempo = 11
1339.                   int[] Resultado = new int[tiempo];
1340.                   double[] X = new Double[tiempo];
1341.
1342.                   double f;
1343.                   for (int i = 1; i <= tiempo; i++)
1344.                   {//Expresión de la Distribución Normal
1345.                       X[i - 1] = Media - (i - tiempo / 2) * Desviacion; //Valores
     de X
1346.                       f = (1 / (Desviacion * Math.Sqrt(2 * Math.PI))) * Math.Pow(M
     ath.E, (-0.5 * Math.Pow(((X[i - 1] - Media) / Desviacion), 2)));
1347.                       Resultado[i - 1] = (int)Math.Round(f * X[i - 1]);
1348.                   }
1349.
1350.                   return Resultado; //Se devuelve el vector con todos los valores
1351.               }
1352.
1353.               int[] DistribucionEnRampa(double Media, int tiempo, bool Sentido)
1354.               {
```

```
1355.                 //Función que obtiene la distribución en Rampa según la media, y
   el sentido de la pendiente
1356.                 int[] Resultado = new int[tiempo + 1];
1357.                 double[] X = new Double[tiempo + 1];
1358.
1359.                 double f;
1360.                 for (int i = 1; i <= (tiempo + 1); i++)
1361.                 {
1362.                     //Se calcula según si es de Pendiente Negativa o Positiva.
1363.                     if (Sentido) f = 2 * (Media / tiempo) * (i - 1); //Ascendent
   e
1364.                     else f = (2 * Media / tiempo) * (tiempo - (i - 1));       //
   Descendente
1365.                     Resultado[i - 1] = (int)Math.Round(f);
1366.                 }
1367.
1368.                 return Resultado;//Se devuelve el vector con todos los valores
1369.             }
1370.             int[] DistribucionConstante(double Media, int tiempo)
1371.             {
1372.                 //Función que obtiene la distribución Constante según la media
1373.                 int[] Resultado = new int[tiempo + 1];
1374.                 for (int i = 1; i <= (tiempo + 1); i++)
1375.                 {
1376.                     Resultado[i - 1] = (int)Media; //Todos los valores son
   iguales
1377.                 }
1378.                 return Resultado;//Se devuelve el vector con todos los valores
1379.             }
1380.
1381.             double Obtener_Distancia(double latI, double lngI, double latF, doub
   le lngF)
1382.             {
1383.                 //Función que obtiene la distancia según coordenadas de un Punto
   Inicial y Final
1384.                 double distancia;
1385.                 //Definición puntos iniciales y Finales
1386.                 inicio = new PointLatLng(latI, lngI);
1387.                 final = new PointLatLng(latF, lngF);
1388.                 List<PointLatLng> puntos = new List<PointLatLng>(); //Lista de
   que contiene los dos puntos
1389.                 puntos.Add(inicio);
1390.                 puntos.Add(final);
1391.                 GMapRoute Ruta_Obtenida; //Se traza la ruta con los puntos
1392.                 try {  Ruta_Obtenida = new GMapRoute(puntos, "Ruta
   ubicacion"); }
1393.                 catch
1394.                 {
1395.                     Ruta_Obtenida = new GMapRoute(puntos, "Ruta
   ubicacion");
1396.                 }
1397.
```

```
1398.                 distancia = Ruta_Obtenida.Distance; //La distancia es una
   característica de la ruta
1399.                 return distancia;
1400.             }
1401.
1402.             double SumaImportancias(double[] Importancias, int tramo )
1403.             {//Función que obtiene la suma de importancias, la cadena (string)
   de importancias
1404.                 //y las entradas principales
1405.                 importancias = "";
1406.                 double sum = 0.0;
1407.                 sum = Importancias.Sum(); //Obtención de la suma de importancias
1408.                 importancias = Importancias[0].ToString();
1409.                 //Obtención de la cadena, con las importancias separadas por
   comas,
1410.                 //para facilitar su posterior lectura
1411.                 for (int i = 1; i < Importancias.Length; i++)
1412.                 {
1413.                     importancias +=  ","+Importancias[i].ToString();
1414.                 }
1415.                 //Definición de las entradas principales
1416.                 if (sum != 0.0) EntradasPrincipales[tramo] = 1;
1417.                 else EntradasPrincipales[tramo] = 0;
1418.                 return sum;
1419.             }
1420.
1421.             void GuardarResultados()
1422.             {
1423.                 //Función para guardar los resultados en el Archivo de Texto
   seleccionado
1424.                 int[] V = new int[iteracion], P = new int[iteracion],
   State = new int[iteracion];
1425.                 double[] T = new double[iteracion],
   Tmenos = new double[iteracion];
1426.                 int Tipo, a=0;
1427.
1428.                 SaveFileDialog save = new SaveFileDialog();
1429.                 save.FileName = "DefaultOutputName.txt";
1430.                 save.Filter = "Text File | *.txt";
1431.                 if (save.ShowDialog() == DialogResult.OK)
1432.                 {//Si la selección de la carpeta y el nombre ha sido correcta...
1433.                     //Se instancia una clase para poder crear y escribir en un
   Archivo de Texto
1434.                     StreamWriter writer = new StreamWriter(save.OpenFile());
1435.                     //Con cada writer.WriteLine se escribe una linia diferente
1436.
1437.                     writer.WriteLine("\"Resultados de la simulación del
   Proyecto: " + NombreProyecto+ "\"");
1438.                     writer.WriteLine("          ");
1439.                     for(int filas = 0; filas < dtSimTram.Rows.Count; filas++)
1440.                     {//Bucle con el que se escriben todos los resultados de cada
   Tramo
1441.                         a = 0;
```

```
1442.                         Tipo = Convert.ToInt16(dtSimTram.Rows[filas][6]); //Vari
   able tipo de semáforos
1443.                         writer.WriteLine("\"\\\\\\\\\\ TRAMO:
   " + Convert.ToString(dtSimTram.Rows[filas][0])+ " \\\\\\\\\"");
1444.                         writer.WriteLine("Iteracion,         Estado,       Vehíc
   ulos,       Peatones,      Tiempo Restante,       Capacidades");
1445.                         for (int i = 0; i < Resultados.Rows.Count; i++) { //Bucl
   e para escribir todos los detalles de un Tramo
1446.                             //Vehículos de todos los Tramos para la iteración i
1447.                             V = Convert.ToString(Resultados.Rows[i][2]).Split(',
   ').Select(x => int.Parse(x)).ToArray();
1448.                             //Peatones de todos los Tramos para la iteración i
1449.                             P = Convert.ToString(Resultados.Rows[i][3]).Split(',
   ').Select(x => int.Parse(x)).ToArray();
1450.                             //Hay que tener en cuenta si hay duplicidad de
   resultados, debido a la diferencia de Tiempos entre Tramos
1451.                             //Tmenos->Tiempos restantes en la iteración anterior
1452.                             if(i>0)Tmenos = Convert.ToString(Resultados.Rows[i-
   1][5]).Split(',').Select(x => double.Parse(x)).ToArray();
1453.                             //T-> Tiempos restantes en la iteración actual
1454.                             T = Convert.ToString(Resultados.Rows[i][5]).Split(',
   ').Select(x => double.Parse(x)).ToArray();
1455.                             //Estados de todos los Tramos para la iteración i
1456.                             State = Convert.ToString(Resultados.Rows[i][1]).Spli
   t(',').Select(x => int.Parse(x)).ToArray();
1457.
1458.                             if ((State[filas] == 1 || Tipo == 4) && Tmenos[filas
   ] != T[filas])
1459.                             {//Si el estado es Activo o el Semáforo es Ámbar,
   además de no tener duplicidad se escribe la siguiente línea
1460.                                 writer.WriteLine("     " + a.ToString() + ",
            1,                " + V[filas].ToString() + ",              " + P[filas]
   .ToString() + ",                " + Math.Round(T[filas], 3).ToString() + ",
          " + Capacidades[filas]);
1461.                                 a++;
1462.                             }
1463.                             else if (Tmenos[filas] != T[filas])
1464.                             {//En cambio si no hay duplicidad y Estado Parado se
   escribe la siguiente línea
1465.                                 writer.WriteLine("     " + a.ToString() + ",
            0,                " + V[filas].ToString() + ",              " + P[filas]
   .ToString() + ",                " + Math.Round(T[filas], 3).ToString() + ",
          " + Capacidades[filas]);
1466.                                 a++;
1467.                             }
1468.                         }
1469.                         writer.WriteLine("          ");//Linia Final
1470.                     }
1471.                     writer.Dispose();
1472.
1473.                     writer.Close();
1474.                     bFinalizar.ForeColor = Color.FromArgb(49, 66, 82); //Se
   desmarca el botón como pulsado
```

```
1475.
1476.                 }
1477.
1478.
1479.
1480.
1481.             }
1482.
1483.             void CrearConfig()
1484.             {//Función para crear la configuración y guardarla en un Archivo de
    Texto
1485.                 //Iniciación de todas las cadenas
1486.                 string[] Nombres = new string[dtSimTram.Rows.Count];
1487.                 Nombres[0] = Convert.ToString(dtSimTram.Rows[0][0]);
1488.                 string LatI = Convert.ToString(dtSimTram.Rows[0][1]);
1489.                 string LongI = Convert.ToString(dtSimTram.Rows[0][2]);
1490.                 string LatF = Convert.ToString(dtSimTram.Rows[0][3]);
1491.                 string LongF = Convert.ToString(dtSimTram.Rows[0][4]);
1492.                 string Vel_KM = Convert.ToString(dtSimTram.Rows[0][5]);
1493.                 string TipoF = Convert.ToString(dtSimTram.Rows[0][6]);
1494.                 string Carriles = Convert.ToString(dtSimTram.Rows[0][7]);
1495.                 string Distancia = Convert.ToString(dtSimTram.Rows[0][8]);
1496.                 string[] Vehiculos = new string[dtSimTram.Rows.Count];
1497.                 Vehiculos[0] = Convert.ToString(dtSimTram.Rows[0][9]);
1498.                 string[] Peatones = new string[dtSimTram.Rows.Count];
1499.                 Peatones[0] = Convert.ToString(dtSimTram.Rows[0][11]);
1500.                 string[] Importancias = new string[dtSimRel.Rows.Count];
1501.                 Importancias[0] = Convert.ToString(dtSimRel.Rows[0][1]);
1502.                 string SumaImportancias = Convert.ToString(dtSimRel.Rows[0][2]);
1503.
1504.                 for(int filas = 1;filas< dtSimTram.Rows.Count; filas++)
1505.                 {//Bucle para poder crear todos los strings con comas para
    habilitar una buena
1506.                     //lectura posterior
1507.                     Nombres[filas] = Convert.ToString(dtSimTram.Rows[filas][0]);
1508.                     LatI += (","+ Convert.ToString(dtSimTram.Rows[filas][1]));
1509.                     LongI += (","+ Convert.ToString(dtSimTram.Rows[filas][2]));
1510.                     LatF += (","+ Convert.ToString(dtSimTram.Rows[filas][3]));
1511.                     LongF += ("," + Convert.ToString(dtSimTram.Rows[filas][4]));
1512.                     Vel_KM += ("," + Convert.ToString(dtSimTram.Rows[filas][5]))
    ;
1513.                     TipoF += ("," + Convert.ToString(dtSimTram.Rows[filas][6]));
1514.                     Carriles += ("," + Convert.ToString(dtSimTram.Rows[filas][7]
    ));
1515.                     Distancia += ("," + Convert.ToString(dtSimTram.Rows[filas][8
    ]));
1516.                     Vehiculos[filas] = Convert.ToString(dtSimTram.Rows[filas][9]
    );
1517.                     Peatones[filas] = Convert.ToString(dtSimTram.Rows[filas][11]
    );
1518.                     Importancias[filas] = Convert.ToString(dtSimRel.Rows[filas][
    1]);
```

```
1519.                     SumaImportancias += ("," + Convert.ToString(dtSimRel.Rows[fi
   las][2]));
1520.                 }
1521.
1522.                 //Uso de la clase para crear y escribir en un Archivo de texto
1523.                 //dir es la variable que almacena la dirección del archivo
   seleccionado
1524.
1525.                 using (StreamWriter myConfig = File.AppendText(dir))
1526.                 {
1527.                     //Se escribe en la primera linea el nombre del proyecto
1528.                     myConfig.WriteLine(NombreProyecto);
1529.                     myConfig.WriteLine("NOMBRES");
1530.                     for (int filas = 0; filas < dtSimTram.Rows.Count; filas++)
1531.                     {//Bucle para escribir todos los nombres
1532.                         myConfig.WriteLine(Nombres[filas]);
1533.                     }
1534.                     //Se escriben las distintas variables
1535.                     myConfig.WriteLine("LATITUDES INICIALES");
1536.                     myConfig.WriteLine(LatI);
1537.                     myConfig.WriteLine("LONGITUDES INICIALES");
1538.                     myConfig.WriteLine(LongI);
1539.                     myConfig.WriteLine("LATITUDES FINALES");
1540.                     myConfig.WriteLine(LatF);
1541.                     myConfig.WriteLine("LONGITUDES FINALES");
1542.                     myConfig.WriteLine(LongF);
1543.                     myConfig.WriteLine("VELOCIDADES KM/H");
1544.                     myConfig.WriteLine(Vel_KM);
1545.                     myConfig.WriteLine("TIPOS");
1546.                     myConfig.WriteLine(TipoF);
1547.                     myConfig.WriteLine("CARRILES");
1548.                     myConfig.WriteLine(Carriles);
1549.                     myConfig.WriteLine("DISTANCIAS");
1550.                     myConfig.WriteLine(Distancia);
1551.                     myConfig.WriteLine("DENSIDAD VEHICULOS");
1552.                     for (int filas = 0; filas < dtSimTram.Rows.Count; filas++)
1553.                     {//Bucle para escribir la configuración del flujo de
   vehículos
1554.                         myConfig.WriteLine(Vehiculos[filas]);
1555.                     }
1556.                     myConfig.WriteLine("DENSIDAD PEATONES");
1557.                     for (int filas = 0; filas < dtSimTram.Rows.Count; filas++)
1558.                     {//Bucle para escribir la configuración del flujo de
   peatones
1559.                         myConfig.WriteLine(Peatones[filas]);
1560.                     }
1561.                     myConfig.WriteLine("IMPORTANCIAS");
1562.                     for (int filas = 0; filas < dtSimTram.Rows.Count; filas++)
1563.                     {//Bucle para escribir la configuración de importancias
1564.                         myConfig.WriteLine(Importancias[filas]);
1565.                     }
1566.                     myConfig.WriteLine("SUMA IMPORTANCIAS");
1567.                     myConfig.WriteLine(SumaImportancias);
```

```
1568.                     myConfig.Close();
1569.                 }
1570.
1571.             }
1572.
1573.             void LeerConfig()
1574.             {//Función para Abrir un Archivo de Texto y obtener la configuración
1575.                 string linea=",";
1576.                 string[] Nombres, Vehiculos, Peatones, Importancias;
1577.                 int RowsCount = 0;
1578.                 double[] LatI, LongI, LatF, LongF, Distancia,Vel_KM,
   SumaImportancias;
1579.                 int[] TipoF, Carriles;
1580.                 //dir es la variable que almacena la dirección del archivo
   seleccionado
1581.                 using (StreamReader myConfig = new StreamReader(dir))
1582.                 {
1583.                     //MessageBox.Show("Leido");
1584.                     //con myConfig.ReadLine se lee una linia
1585.                     linea = myConfig.ReadLine();
1586.                     while ((linea) != null && linea != "LATITUDES
   INICIALES") //Leer linea por linea
1587.                     {//Bucle para poder conocer en que linea empieza a leerse la
   configuración
1588.                         linea = myConfig.ReadLine();
1589.                         RowsCount++;
1590.
1591.                     }
1592.                     RowsCount -=2;
1593.                     Nombres = new string[RowsCount];
1594.                     Vehiculos = new string[RowsCount];
1595.                     Peatones = new string[RowsCount];
1596.                     Importancias = new string[RowsCount];
1597.
1598.                     myConfig.Close();
1599.                 }
1600.                 //dir es la variable que almacena la dirección del archivo
   seleccionado
1601.                 using (StreamReader myConfig = new StreamReader(dir))
1602.                 { //Se vuelve a abrir el archivo
1603.                     NombreProyecto = myConfig.ReadLine();//Se lee el nombre del
   proyecto
1604.                     linea = myConfig.ReadLine();
1605.                     if (linea == "NOMBRES")//Si la linea obtenida es Nombres,
   empieza la lectura
1606.                     {
1607.                         linea = myConfig.ReadLine();
1608.                         for (int filas = 0; filas < RowsCount; filas++)
1609.                         {//Bucle para almacenar los nombres de los tramos
1610.                             Nombres[filas] = linea;
1611.                             linea = myConfig.ReadLine(); // En la ultima
   iteracion ->linia = LATITUDES INICIALES
1612.                         }
```

```
1613.                           //MessageBox.Show(RowsCount.ToString());
1614.                           LatI = myConfig.ReadLine().Split(',').Select(x => double
   .Parse(x)).ToArray();
1615.                           linea = myConfig.ReadLine(); // línea = LONGITUDES
   INICIALES
1616.                           LongI = myConfig.ReadLine().Split(',').Select(x => doubl
   e.Parse(x)).ToArray();
1617.                           linea = myConfig.ReadLine(); // línea = LATITUDES
   FINALES
1618.                           LatF = myConfig.ReadLine().Split(',').Select(x => double
   .Parse(x)).ToArray();
1619.                           linea = myConfig.ReadLine(); // línea = LONGITUDES
   FINALES
1620.                           LongF = myConfig.ReadLine().Split(',').Select(x => doubl
   e.Parse(x)).ToArray();
1621.                           linea = myConfig.ReadLine(); // línea = VELOCIDADES KM/H
1622.                           Vel_KM = myConfig.ReadLine().Split(',').Select(x => doub
   le.Parse(x)).ToArray();
1623.                           linea = myConfig.ReadLine(); // línea = TIPOS
1624.                           TipoF = myConfig.ReadLine().Split(',').Select(x => int.P
   arse(x)).ToArray();
1625.                           linea = myConfig.ReadLine(); // línea = CARRILES
1626.                           Carriles = myConfig.ReadLine().Split(',').Select(x => in
   t.Parse(x)).ToArray();
1627.                           linea = myConfig.ReadLine(); // línea = DISTANCIAS
1628.                           Distancia = myConfig.ReadLine().Split(',').Select(x => d
   ouble.Parse(x)).ToArray();
1629.                           linea = myConfig.ReadLine(); // línea = DENSIDAD
   VEHICULOS
1630.
1631.
1632.                           for (int filas = 0; filas < RowsCount; filas++)
1633.                           {//Se guardan la configuración de flujo de vehículos
1634.                               Vehiculos[filas] = myConfig.ReadLine();
1635.                           }
1636.
1637.                           linea = myConfig.ReadLine(); // linia = DENSIDAD
   PEATONES
1638.                           for (int filas = 0; filas < RowsCount; filas++)
1639.                           {//Se guarda la configuración de flujo de peatones
1640.                               Peatones[filas] = myConfig.ReadLine();
1641.                           }
1642.                           linea = myConfig.ReadLine(); // linia =
   IMPORTANCIAS
1643.                           for (int filas = 0; filas < RowsCount; filas++)
1644.                           {//Se guardan las importancias
1645.                               Importancias[filas] = myConfig.ReadLine();
1646.                           }
1647.
1648.                           linea = myConfig.ReadLine(); // línea = SUMA
   IMPORTANCIAS
1649.                           SumaImportancias = myConfig.ReadLine().Split(',').Select
   (x => double.Parse(x)).ToArray();
```

```
1650.                         myConfig.Close();
1651.
1652.
1653.                         dtSimTram.Clear();//Se limpia la Tabla de Datos 1
1654.                         dtSimRel.Clear();//Limpieza de la Tabla de Datos 2
1655.                         for (int filas = 0; filas < RowsCount; filas++)
1656.                         {//Bucle para llenar las Tablas de Datos 1 y 2 con los
   valores obtenidos
1657.                             dtSimTram.Rows.Add(Nombres[filas], LatI[filas],
   LongI[filas], LatF[filas], LongF[filas], Vel_KM[filas], TipoF[filas],
   Carriles[filas], Distancia[filas], Vehiculos[filas], false, Peatones[filas]);
1658.                             dtSimRel.Rows.Add(Nombres[filas],
   Importancias[filas], SumaImportancias[filas]);
1659.                         }
1660.                     }else {//Si la linea no es "Nombres"
1661.                         MessageBox.Show(linea);
1662.                         NombreProyecto = null;
1663.                     }
1664.                 }
1665.
1666.             }
1667.
1668.
1669.             GMarkerGoogle
   Marcadores(bool Estado,int Tipo,double LatI,double LngI,double LatF,double LngF,i
   nt CiclosRestantes,bool Alternancia)
1670.             {//Función que varía el color de los marcadores en función del tipo
   de semáforo del Tramo
1671.                 switch (Tipo)
1672.                 {
1673.                     case 0: //Nulo
1674.                         marker = new GMarkerGoogle(new PointLatLng((LatF + LatI)
    * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.white_small);
1675.                         break;
1676.                     case 1: //Verde y Rojo
1677.                         if (Estado) //Verde
1678.                         {
1679.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.green_small);
1680.                         }
1681.                         else//Rojo
1682.                         {
1683.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.red_small);
1684.                         }
1685.                         break;
1686.                     case 2: //Verde, Rojo y Ámbar
1687.                         if (Estado && CiclosRestantes > MinAmbar) //Verde
1688.                         {
1689.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.green_small);
1690.                         }
1691.                         else if (Estado && CiclosRestantes <= MinAmbar) //Ámbar
```

```
1692.                         {
1693.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.yellow_small);
1694.                         }
1695.                         else//Rojo
1696.                         {
1697.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.red_small);
1698.                         }
1699.                         break;
1700.                     case 3: //Ámbar y Rojo
1701.                         if (Estado) //Ámbar
1702.                         {
1703.                             if (Alternancia) // Par -> Ámbar
1704.                             {
1705.                                 marker = new GMarkerGoogle(new PointLatLng((LatF
    + LatI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.yellow_small);
1706.                             }
1707.                             else //Impar -> Negro
1708.                             {
1709.                                 marker = new GMarkerGoogle(new PointLatLng((LatF
    + LatI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.black_small);
1710.                             }
1711.                         }
1712.                         else // Rojo
1713.                         {
1714.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.red_small);
1715.                         }
1716.                         break;
1717.                     case 4: //Ámbar
1718.                         if (iteracion % 2 == 0) // Par -> Ámbar
1719.                         {
1720.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.yellow_small);
1721.                         }
1722.                         else //Impar -> Negro
1723.                         {
1724.                             marker = new GMarkerGoogle(new PointLatLng((LatF + L
   atI) * 0.5, (LngF + LngI) * 0.5), GMarkerGoogleType.black_small);
1725.                         }
1726.                         break;
1727.                 }
1728.                 return marker;
1729.             }
1730.         }
1731.     }
```

Anexo II: Código Clase Algoritmo

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;//Permite extraer vectores a partir de cadenas
4. using System.Text;
5. using System.Threading.Tasks;
6. using System.Data;//Permite trabajar con Bases de Datos
7. using System.Windows.Forms;
8.
9. namespace ProjecteTFG
10.     {
11.         class Algoritmo
12.         {
13.             int[] UltimosValores;
14.             int[] Embotellamientos;//Variable de vector de enteros para guardar
    embotellamientos
15.             public DataTable Operaciones(DataTable Tram,  DataTable Rel,
    DataTable Sim, DataGridView
    Parrilla, int iteracion, double Ts, int[] EntradasP, double[] Distancias)
16.             {//Función que obtiene una nueva Tabla de Datos 3 actualizada
17.                 //Iniciación de las variables
18.                 int[] ENTRADAS = new int[EntradasP.Length];//Vector entradas
    principales
19.                 //Vector para trabajar con embotellamientos en cada Tramo
20.                 Embotellamientos = new int[Tram.Rows.Count];
21.                 double TiempoR, Tiempo;
22.                 ENTRADAS = EntradasP;
23.                 Parrilla.DataSource = Sim;
24.                 UltimosValores = new int[Parrilla.RowCount];
25.                 double[] Tiempos = new double[Parrilla.RowCount];
26.
27.                 if (iteracion == -1)
28.                 {//Si iteración vale -1, significa que es la primera vez que se
    llama esta función
29.                     int[] Celda3;
30.                     for(int filas = 0; filas< Tram.Rows.Count; filas++)
31.                     {
32.                         //Bucle para llenar la Tabla de Datos
33.                         //Vector de vehículos de cada tramo, con una longitud
    igual a
34.                         //cuantos vehículos quepan (5metros longitud media de
    los vehículos)
35.
36.                         Celda3 = new Int32[Convert.ToInt32(Distancias[filas] / 5
    )];
37.                         Celda3 = Inicializar(Celda3);//Todos los valores del
    vector se ponen a 0
38.                         Tiempo = CalcularTiempo(Convert.ToDouble(Tram.Rows[filas
    ][5])); //Tiempo de iteración de cada Tramo
39.                         //Se añaden los valores a la Tabla de Datos
40.                         Sim.Rows.Add(false,0,
    Convert.ToInt32(Convert.ToDouble(Tram.Rows[filas][8]) * 1000.0 / 5),
    CrearString(Celda3),0,0,0,0,0,Tiempo,Tiempo,true,0,0);
41.                         Parrilla.Rows[filas].Cells[2].Value = Convert.ToInt32(15
    / CalcularTiempo(Convert.ToDouble(Tram.Rows[filas][5])));
42.                         Embotellamientos[filas] = 0;
43.                     }
44.                 }
45.                 else
46.                 {//Si la simulación ya ha empezado...
```

```
47.
48.                         //int[,] Incompatibilidades = new int[Parrilla.RowCount,
    Parrilla.RowCount];
49.                         for (int filas = 0; filas < Parrilla.RowCount; filas++)
50.                         {//Se obtienen los vehículos pueden abandonar su Tramo
51.                             UltimosValores[filas] = Convert.ToInt32(Sim.Rows[filas][
    8]);
52.                             //Obtención de los tiempos de iteración de cada Tramo
53.                             Tiempos[filas] = Convert.ToDouble(Sim.Rows[filas][10]);
54.                         }
55.
56.                         for (int filas = 0; filas < Parrilla.RowCount; filas++)
57.                         {
58.                             //Obtención de los valores de la Tabla de Datos 3
59.                             Boolean Celda0 = Convert.ToBoolean(Sim.Rows[filas][0]);
60.                             int Celda1 = Convert.ToInt16(Sim.Rows[filas][1]),
    Celda2 = Convert.ToInt16(Sim.Rows[filas][2]);
61.                             int[] Celda3 = Sim.Rows[filas][3].ToString().Split(',').
    Select(x => int.Parse(x)).ToArray(); //Extrae el vector int guardado
    convirtiéndolo desde String
62.                             int Celda4 = Convert.ToInt16(Sim.Rows[filas][4]),
    Celda5 = Convert.ToInt16(Sim.Rows[filas][5]);
63.                             int Celda6 = Convert.ToInt32(Sim.Rows[filas][6]),
    Celda7 = Convert.ToInt32(Sim.Rows[filas][7]);
64.                             int Tipo = Convert.ToInt32(Tram.Rows[filas][6]),Celda12
    = Convert.ToInt32(Sim.Rows[filas][12]);
65.                             Tiempo = Convert.ToDouble(Sim.Rows[filas][10]);
66.                             TiempoR = Convert.ToDouble(Sim.Rows[filas][9]);//Tiempo
    restante para finalizar la iteración actual
67.
68.                             if (TiempoR <= Tiempos.Min())
69.                             {//Si la iteración actual llega a su fin...
70.
71.                                 if (((Celda2 >= 1) && (Celda0)) || Tipo == 0 || Tipo
     == 4) //Si todavía no tiene que cambiar de estado
72.                                 { //Continuidad en Verde, si el semáforo es nulo en
    el tramo, el tránsito fluye constantemente
73.                                     Celda1++; //Incremento de ciclos en estado
    actual
74.                                     Celda12++; //Incremento en la iteración propia
    del Tramo
75.                                     if(Tipo!=4 && Tipo !=0)Celda2--; //Si el
    semáforo no es nulo o ámbar, se reducen los ciclos restantes
76.                                     Celda4 -= UltimosValores[filas]; //Restarle el
    último valor que son los vehículos que se van
77.                                     //Desplazamiento del vector de vehículos debido
    al desplazamiento de estos
78.                                     Celda3[Celda3.Length-1] = 0;
79.                                     Celda3 = DesplazarInt(Celda3, "Verde", Celda4,
    Convert.ToInt32(Tram.Rows[filas][7]));
80.                                     //La primera celda la ocupan los vehículos
    entrantes
81.                                     Celda3[0] = CalculoEntradaV(filas, Tram, Rel,
    Sim, Parrilla, Celda1, EntradasP[filas]); //Cálculo del nuevo flujo de vehíulos
82.                                     Celda4 = Celda3.Sum();//Sumar los vehículos
    entrantes al total de vehículos en el Tramo
83.                                     //Sumar los nuevos peatones que se acumulan
84.                                     Celda5 += CalcularPeatones(filas, Tram, Sim,
    Parrilla, Celda12); //Sumar los nuevos peatones que se acumulan
85.                                 }
86.                                 else if ((Celda2 == 0) && (Celda0)) //Puesta en Rojo
```

```
87.                                 {//Si no hay ciclos restantes y el estado actual es
   activo...
88.
89.                                     Celda0 = !Celda0;//Cambiar el estado
90.                                     Celda12++;//Incremento en la iteración propia
   del Tramo
91.                                     //Desplazamiento del vector de vehículos debido
   al desplazamiento de estos últimos
92.                                     Celda3[Celda3.Length - 1] = 0;
93.                                     Celda3 = DesplazarInt(Celda3, "Verde", Celda4,
   Convert.ToInt32(Tram.Rows[filas][7]));
94.                                     //Cálculo de los vehículos entrantes
95.                                     Celda3[0] = CalculoEntradaV(filas, Tram, Rel,
   Sim, Parrilla, Celda12, EntradasP[filas]);
96.                                     //Actualización del valor total de los vehículos
   en el Tramo
97.                                     Celda4 = Celda3.Sum();
98.                                     //Sumar los nuevos peatones que se acumulan
99.                                     Celda5 += CalcularPeatones(filas, Tram, Sim,
   Parrilla, Celda12);
100.                                    //Cálculo de los ciclos restantes, ahora en
   estado parado
101.                                    Celda2 = CalculoTiempoAR(filas, Rel, Sim, Tram,
   Parrilla); //0 para pasar a Rojo
102.                                    Celda1 = 0;//Ciclos en estado actual ->0
103.
104.                                    //Celdas que se almacenan para variar el
   rendimiento
105.                                    Celda6 = Celda4;
106.                                }
107.                                else if ((Celda2 >= 1) && (Celda0 == false)) //Conti
   nuidad en Rojo
108.                                {//Si el Estado es Parado y hay ciclos restantes...
109.
110.                                    Celda12++;//Incremento en la iteración propia
   del Tramo
111.                                    Celda1++;//Incremento de ciclos en estado actual
112.                                    Celda2--;// se reducen los ciclos restantes
113.                                    //Desplazamiento del vector de vehículos debido
   al desplazamiento de estos últimos
114.                                    Celda3 = DesplazarInt(Celda3, "Rojo", Celda4,
   Convert.ToInt32(Tram.Rows[filas][7]));
115.                                    //Cálculo de los vehículos entrantes
116.                                    Celda3[0] += CalculoEntradaV(filas, Tram, Rel,
   Sim, Parrilla, Celda12, EntradasP[filas]);
117.                                    //Actualización del valor total de los vehículos
   en el Tramo
118.                                    Celda4 = Celda3.Sum();
119.                                    //No hay peatones
120.                                    Celda5 = 0;
121.                                }
122.                                else if ((Celda2 == 0) && (!Celda0)) //Puesta en
   Verde
123.                                {//Si el Estado es Parado y ya no quedan ciclos...
124.
125.                                    Celda12++;//Incremento en la iteración propia
   del Tramo
126.                                    //Desplazamiento del vector de vehículos debido
   al desplazamiento de estos últimos
127.                                    Celda3 = DesplazarInt(Celda3, "Rojo", Celda4,
   Convert.ToInt32(Tram.Rows[filas][7]));
```

```
128.                                    //Cálculo de los vehículos entrantes
129.                                    Celda3[0] += CalculoEntradaV(filas, Tram, Rel,
    Sim, Parrilla, Celda12, EntradasP[filas]); //Cálculo del nuevo flujo de vehículos
130.                                    //Actualización del valor total de los vehículos
    en el Tramo
131.                                    Celda4= Celda3.Sum();
132.
133.                                    //Cálculo de la nueva cantidad de ciclos en
    Estado Activo
134.                                    Celda2 = CalculoTiempoAV(filas, Rel, Sim, Tram,
    Parrilla); //0 para pasar a Rojo
135.                                    if (Celda2 < 0)
136.                                    {//Si el valor de Celda2 obtenido es negativo,
    hay que permanecer en estado parado
137.                                        Celda2 = Math.Abs(Celda2);
138.                                        Celda0 = false;
139.                                    }
140.                                    else Celda0 = true;
141.                                    Celda1 = 0;//Ciclos en Estado Actual = 0
142.                                    Celda5 = 0;//Peatones = 0
143.                                    Celda7 = Celda4;//Guardado de valores para el
    rendimiento
144.                                }
145.                                //Actualización del tiempo restante de la iteración
    del Tramo
146.                                TiempoR = Tiempo - (Tiempos.Min() - TiempoR);
147.                                //Actualización de la Tabla de Datos 3
148.                                Sim.Rows[filas][0] = Celda0;
149.                                Sim.Rows[filas][1] = Celda1;
150.                                Sim.Rows[filas][2] = Celda2;
151.                                Sim.Rows[filas][3] = CrearString(Celda3);
152.                                Sim.Rows[filas][4] = Celda4;
153.                                Sim.Rows[filas][5] = Celda5;
154.                                Sim.Rows[filas][6] = Celda6;
155.                                Sim.Rows[filas][7] = Celda7;
156.                                Sim.Rows[filas][9] = TiempoR;
157.                                Sim.Rows[filas][12] = Celda12;
158.                                //Si se ha seleccionado ámbar, que este alterne-
    >Variación de Alternancia
159.                                if (Convert.ToInt16(Tram.Rows[filas][6]) >= 3) Sim.R
    ows[filas][11] = !Convert.ToBoolean(Sim.Rows[filas][11]);
160.
161.                            }
162.                            else
163.                            {//Si el tiempo restante era superior al tiempo que
    transcurre, el restante se reduce
164.                                TiempoR -= Tiempos.Min();
165.                                Sim.Rows[filas][9] = TiempoR;
166.                            }
167.                        }
168.                        for (int filas = 0; filas < Parrilla.RowCount; filas++)
169.                        {
170.                            //Actualización de la Celda 3-> Vector de vehículos
171.                            int[] Celda3 = Sim.Rows[filas][3].ToString().Split(',').
    Select(x => int.Parse(x)).ToArray(); //Extrae el vector int guardado
    convirtiéndolo desde String
172.                            Sim.Rows[filas][8] = Celda3.Last();
173.                        }
174.
175.                    }
```

```
176.                    //Extrae el vector int guardado
    conviertienTram.Rows[Tramo][8]dolo desde String
177.                    return Sim; //Devuelve la nueva Tabla de Datos 3
178.                }
179.
180.                int CalculoTiempoAR(int Tramo, DataTable Rel, DataTable Sim,
    DataTable Tram, DataGridView Parrilla)
181.                {
182.                    //Función que calcula los ciclos que va a estar Tramo en estado
    Parado
183.                    int Celda4, Celda6,Periodos;
184.                    double Rendimiento;
185.                    //Obtención de la distancia
186.                    double distancia = Convert.ToDouble(Tram.Rows[Tramo][8])*1000.0;
187.                    int Tmin;
188.                    //Los ciclos parados deben ocupar como mínimo 20seg a variar
    según el rendimiento
189.                    Tmin = Convert.ToInt32(20 / CalcularTiempo(Convert.ToDouble(Tram
    .Rows[Tramo][5])));
190.                    Parrilla.DataSource = Sim;
191.                    Celda4 = Convert.ToInt32(Sim.Rows[Tramo][4]);
192.                    Celda6 = Convert.ToInt32(Sim.Rows[Tramo][6]);
193.                    if (Celda4 == 0) Rendimiento = 1.0;
194.                    else Rendimiento = Convert.ToDouble(Celda6 / Celda4); //Cálculo
    del Rendimiento
195.                    //El rendimiento se basa en que si hay más vehículos o no
    respecto a estados anteriores
196.                    //El rendimiento afecta a la cantidad de ciclos que devuelve la
    función
197.                    //El máximo tiempo en Rojo son 30 segundos.
198.                    Periodos= Convert.ToInt32((Convert.ToInt16(Parrilla.Rows[Tramo].
    Cells[1].Value) * Rendimiento));
199.                    if (Periodos * CalcularTiempo(Convert.ToDouble(Tram.Rows[Tramo][
    5])) > 30)
200.                        Periodos = Convert.ToInt32(30 / CalcularTiempo(Convert.ToDou
    ble(Tram.Rows[Tramo][5])));
201.                    else if (Periodos < Tmin) Periodos = Tmin;
202.                    return Periodos;
203.                }
204.
205.                int CalculoTiempoAV(int Tramo, DataTable Rel, DataTable Sim,
    DataTable Tram, DataGridView Parrilla)
206.                {
207.                    //Función que calcula los ciclos que va a estar Tramo en estado
    Activo
208.                    int Celda4, Celda7;
209.                    double Rendimiento;
210.
211.                    Parrilla.DataSource = Rel;
212.                    int[] Resultado = new int[Parrilla.RowCount];
213.                    //Determina si la activación del Tramo es posible
214.                    Resultado = ObtenerIncompatibilidades(Rel,Sim, Tramo);
215.                    //Obtención de la distancia
216.                    double distancia = Convert.ToDouble(Tram.Rows[Tramo][8]) * 1000.
    0;
217.                    int Periodos;
218.                    int Tmin;
219.                    //Los ciclos deben ocupar como mínimo 30seg a variar según el
    rendimiento
220.                    Tmin = Convert.ToInt32(30 / CalcularTiempo(Convert.ToDouble(Tram
    .Rows[Tramo][5])));
```

```
221.                 int Dec= Decision(Resultado,Sim,Tram,Tramo);
222.                 Celda4 = Convert.ToInt32(Sim.Rows[Tramo][4]);
223.                 Celda7 = Convert.ToInt32(Sim.Rows[Tramo][7]);
224.
225.                 if (Dec<0)
226.                 {
227.                     return Dec;//En este caso se espera
228.                 }
229.                 else if(Dec == 1 || Resultado.Sum()==0)
230.                 {//Los tiempos en activo deben de ser de como máximo 45 seg.
231.                     if (Celda4 == 0) Rendimiento = 1.0;
232.                     else Rendimiento = Convert.ToDouble(Celda7 / Celda4);
233.                     Periodos = Convert.ToInt32(Convert.ToInt16(Sim.Rows[Tramo][1
    ]) * Rendimiento);
234.                     if (Periodos * CalcularTiempo(Convert.ToDouble(Tram.Rows[Tra
    mo][5])) > 45)
235.                         Periodos = Convert.ToInt32(45 / CalcularTiempo(Convert.T
    oDouble(Tram.Rows[Tramo][5])));
236.                     else Periodos = Tmin;
237.                     return Periodos;
238.
239.                 }
240.
241.                 Celda4 = Convert.ToInt32(Sim.Rows[Tramo][4]);
242.                 Celda7 = Convert.ToInt32(Sim.Rows[Tramo][7]);
243.                 if (Celda4 == 0) Rendimiento = 1.0;
244.                 else Rendimiento = Convert.ToDouble(Celda7 / Celda4);
245.                 Periodos = Convert.ToInt32(Convert.ToInt16(Sim.Rows[Tramo][1]) *
     Rendimiento);
246.                 if (Periodos * CalcularTiempo(Convert.ToDouble(Tram.Rows[Tramo][
    5])) > 45)
247.                     Periodos = Convert.ToInt32(45 / CalcularTiempo(Convert.ToDou
    ble(Tram.Rows[Tramo][5])));
248.                 else Periodos = Tmin;
249.                 return Periodos;
250.             }
251.
252.             int[] ObtenerIncompatibilidades(DataTable Rel, DataTable
    Sim, int Tramo)
253.             {
254.                 //Esta función pregunta si la activación de Tramo molesta a
    otros
255.                 double[] ModeloVehículos = new double[Rel.Rows.Count];
256.                 int[] Resultado = new int[Rel.Rows.Count];
257.                 for (int filas = 0; filas < Rel.Rows.Count; filas++)
258.                 {
259.                     Resultado[filas] = 0;
260.                 }
261.
262.                     for (int filas = 0; filas < Rel.Rows.Count; filas++)
263.                 {//Extrae el vector importancias double guardado convirtiéndolo
    desde String
264.                     ModeloVehículos = Rel.Rows[filas][1].ToString().Split(',').S
    elect(x => double.Parse(x)).ToArray();
265.
266.                     //Se marcan en 1 los tramos que afectan a Tramo y en 0 a los
    que no.
267.                     if ((Tramo != filas) && (ModeloVehículos[Tramo] != 100.0) &&
     (ModeloVehículos[Tramo] > 0))
268.                     {
269.                         for (int i = 0; i < Rel.Rows.Count; i++)
```

```
270.                             {
271.                                 if ((ModeloVehículos[i] != 100.0)&&(ModeloVehículos[
    i] > 0))
272.                                     Resultado[i] = 1;
273.                                 else
274.                                 {
275.                                     Resultado[i] = 0;
276.                                 }
277.                             }
278.                         }
279.                     }
280.
281.                     return Resultado;
282.                 }
283.
284.                 int Decision(int[] Resultado,DataTable Sim, DataTable
    Tram,int Tramo)
285.                 {
286.
287.                     double[] Retorno = new double[Sim.Rows.Count];
288.                     int Capacidad, Decision=1, nCarriles, CR = 0,V = 0;
289.                     bool Estado;
290.                     double TiempoN, TiempoS;
291.                     //Aquí se decide si Tramo se pone en verde o si espera en rojo.
292.                     for (int filas = 0; filas < Resultado.Length; filas++)
293.                     {
294.                         nCarriles = Convert.ToInt32(Tram.Rows[filas][7]);
295.     //La función indica que tramo está más lleno, con lo cual, con más vehículos
    mayor probabilidad en caso de indecisión.
296.                         Capacidad = Convert.ToInt32(Convert.ToDouble(Tram.Rows[filas
    ][8]) * 1000.0 * Convert.ToInt32(Tram.Rows[0][7]) / 5); //Celda 8 distancia,
    Celda 7 Carriles
297.                         Estado = Convert.ToBoolean(Sim.Rows[filas][0]);
298.                         V = Convert.ToInt32(Sim.Rows[filas][4]);
299.                         if (Resultado[filas] == 1 && Estado == false)
300.                         {
301.                             //Cálculo de coeficientes
302.                             try { Retorno[filas] = Convert.ToDouble(V / Capacidad);
    }
303.                             catch {
304.                             }
305.                         }
306.                         else if(Resultado[filas] == 1 && Estado == true)
307.                         {
308.                             //Si hay tramos que interrumpen
309.                             Decision = 0;
310.                         }
311.                     }
312.
313.
314.                     if (Retorno[Tramo] == Retorno.Max() && Decision == 1)
315.                     {
316.                         //Si es el máximo se activa
317.                         return 1;
318.                     }
319.                     else {
320.
321.                         for (int filas = 0; filas < Resultado.Length; filas++)
322.                         {
323.                             Estado = Convert.ToBoolean(Sim.Rows[filas][0]);
324.                             CR = Convert.ToInt32(Sim.Rows[filas][2]);
```

```
325.                         if (Estado == true && Resultado[filas]== 1)
326.                         {
327.                             TiempoN = CalcularTiempo(Convert.ToDouble(Tram.Rows[
    Tramo][5]));
328.                             TiempoS = CalcularTiempo(Convert.ToDouble(Tram.Rows[
    filas][5]));
329.                             //Un ciclo de tramo filas dura diferente a un ciclo
    de Tramo de estudio
330.                             CR = (int)((double)CR * TiempoS / TiempoN);
331.                             //Si hay tramos que interrumpen se espera
332.                             return -CR;
333.                         }
334.                     }
335.
336.                 }
337.
338.                 return 0;//Se devuelve las posibilidades de cada tramo para
    ponerse en verde
339.
340.             }
341.
342.             int[] DesplazarInt (int[] Vector,String Color, int Vehiculos, int nC
    arriles)
343.             {
344.                 //La función retorna el vector de vehículos desplazado en
    función de su estado
345.                 int falta;
346.
347.                 if (Color == "Rojo")
348.                 {
349.                     for (int i = (Vector.Length - 1); i > 0; i--)
350.                     {
351.                         if (Vector[i] < nCarriles)
352.                         {
353.                             falta = nCarriles - Vector[i];
354.                             if (falta > Vector[i - 1])
355.                             {
356.                                 Vector[i] += Vector[i - 1];
357.                                 Vector[i - 1] = 0;
358.                             }
359.                             else if (falta <= Vector[i - 1])
360.                             {
361.                                 Vector[i] = nCarriles;
362.                                 Vector[i - 1] -= falta;
363.                             }
364.                         }
365.                     }
366.                 }else if(Color == "Verde")
367.                 {
368.                     for (int i = (Vector.Length - 1); i > 0; i--)
369.                     {
370.                         Vector[i] = Vector[i - 1];
371.                     }
372.                 }
373.
374.
375.                 return Vector;
376.             }
377.
378.             public double CalcularTiempo(double Vel_KMH)
379.             {//Se calcula el tiempo que dura una iteración en cada Tramo
```

```
380.                 //este tiempo se trata del valor que tarda un vehículo en
   recorrer 5 metros con
381.                 // la velocidad media aportada
382.                 double tiempo;
383.                 tiempo = Math.Pow(Vel_KMH * 1000 / (5 * 3600), -1.0);
384.                 return tiempo;
385.             }
386.
387.             int CalculoEntradaV(int Tramo, DataTable Tram, DataTable Rel,
   DataTable Sim, DataGridView Parrilla,int iteracion,int EntradasP)
388.             {
389.                 //Cálculo de vehículos entrnates
390.                 int[] Y = new int[11];
391.                 int Sobrante = 0, Salida=0;
392.                 int nCarriles = Convert.ToInt32(Tram.Rows[Tramo][7]);
393.
394.                 if (EntradasP == 1) //Si es una entrada Principal
395.                 {
396.                     //Se obtienen los vehículos entrantes según la distribución
   escogida
397.                     double[] ModeloVehículos = new double[3];
398.                     Parrilla.DataSource = Tram;
399.                     //Extrae el vector double guardado convirtiéndolo desde
   String
400.                     ModeloVehículos = Tram.Rows[Tramo][9].ToString().Split(',').
   Select(x => double.Parse(x)).ToArray();
401.                     switch (ModeloVehículos[0])
402.                     {
403.                         case 1.0:
404.                             Y = DistribucionNormal(ModeloVehículos[1],
   ModeloVehículos[2],10);
405.                             break;
406.                         case 2.0:
407.                             Y = DistribucionEnRampa(ModeloVehículos[1], 10,false
   );
408.                             break;
409.                         case 2.5:
410.                             Y = DistribucionEnRampa(ModeloVehículos[1], 10, true
   );
411.                             break;
412.                         case 3.0:
413.                              Y = DistribucionConstante(ModeloVehículos[1], 10);
414.                              break;
415.                         default: break;
416.                     }
417.                     //Se devuelve el valor de la celda correspondiente a la
   iteración actual
418.                     Salida = Convert.ToInt32(Y[iteracion % 11]);
419.                 }
420.                 else
421.                 {
422.                     //Si el Tramo no es una entrada principal
423.                     int y=0;
424.                     double[] ModeloVehículos = new double[Rel.Rows.Count];
425.                     Parrilla.DataSource = Rel;
426.                     //En este caso, ModeloVehiculos guarda las importancias de
   las salidas respecto a la entrada Tramo
427.                     //Extrae el vector double guardado convirtiéndolo desde
   String
428.                     ModeloVehículos = Rel.Rows[Tramo][1].ToString().Split(',').S
   elect(x => double.Parse(x)).ToArray();
```

```
429.                     Parrilla.DataSource = Sim;
430.                     for (int i = 0; i < ModeloVehículos.Length; i++)
431.                     {
432.                         //Los vehículos entrantes se calculan con los estados,
    los vehículos salientes de otros tramos y las importancias
433.                         if((Convert.ToBoolean(Sim.Rows[i][0])==true) || (Convert
    .ToBoolean(Sim.Rows[i][0]) == false && (Convert.ToInt32(Sim.Rows[i][1])==0)))
434.                         {
435.                             y += ObtencionEntradas(Tramo,i, ModeloVehículos);
436.                             //y = ((ModeloVehículos[i] *
    UltimosValores[i])/100.0)+y;
437.                         }
438.
439.                     }
440.
441.                     Salida = Convert.ToInt32(y);
442.
443.                 }
444.
445.                 //Además hay que tener en cuenta el posible enbotellamiento de
    vehículos que quieran entrar
446.                 if (Salida < nCarriles) //Si no hay embotellamiento, salen los
    calculados y los que puedan estar parados
447.                 {
448.                     Sobrante = nCarriles - Salida;
449.                     if (Embotellamientos[Tramo] > Sobrante)
450.                     {//Si hay demasiado embotellamiento
451.                         Salida = nCarriles;
452.                         Embotellamientos[Tramo] -= Sobrante;
453.                     }
454.                     else {//Si no hay demasiado embotellamiento
455.                         Salida += Embotellamientos[Tramo];
456.                         Embotellamientos[Tramo] = 0;
457.                     }
458.                 }
459.                 else
460.                 { // Si hay embotellamiento, este se acumula
461.                     Sobrante = Salida - nCarriles;
462.                     Salida = nCarriles;
463.                     Embotellamientos[Tramo] += Sobrante;
464.                 }
465.                 return Salida;//Se devuelven los vehículos entrantes
466.
467.
468.             }
469.
470.             int CalcularPeatones(int Tramo, DataTable Tram, DataTable Sim,
    DataGridView Parrilla,int iteracion)
471.             {
472.                 //Función que calcula los peatones
473.                 double[] ModeloVehículos;
474.                 int[] Y = new int[11];
475.                 Parrilla.DataSource = Tram;
476.                 //Extrae el vector double guardado conviertiendolo desde String
477.                 ModeloVehículos = Parrilla.Rows[Tramo].Cells[11].Value.ToString(
    ).Split(',').Select(x => double.Parse(x)).ToArray();
478.                 switch (ModeloVehículos[0])
479.                 {
480.                     case 1.0:
481.                         Y = DistribucionNormal(ModeloVehículos[1],
    ModeloVehículos[2], 10);
```

```
482.                         break;
483.                     case 2.0:
484.                         Y = DistribucionEnRampa(ModeloVehículos[1], 10, false);
485.                         break;
486.                     case 2.5:
487.                         Y = DistribucionEnRampa(ModeloVehículos[1], 10, true);
488.                         break;
489.                     case 3.0:
490.                         Y = DistribucionConstante(ModeloVehículos[1], 10);
491.                         break;
492.                     default: break;
493.                 }
494.                 Parrilla.DataSource = Sim;
495.                 //El valor depende de la distribución escogida
496.                 return Convert.ToInt32(Y[iteracion % 11]);
497.             }
498.
499.             int[] DistribucionNormal(double Media, double Desviacion, int tiempo
    )
500.             {
501.                 //Función de la Expresión de la Distribución Normal según la
    media y la desviación
502.                 tiempo++; //Para ajustar al tiempo real 10 -> tiempo = 11
503.                 int[] Resultado = new int[tiempo];
504.                 double[] X = new Double[tiempo];
505.
506.                 double f;
507.                 for (int i = 1; i <= tiempo; i++)
508.                 {//Expresión de la Distribución Normal
509.                     X[i - 1] = Media - (i - tiempo / 2) * Desviacion; //Valores
    de X
510.                     f = (1 / (Desviacion * Math.Sqrt(2 * Math.PI))) * Math.Pow(M
    ath.E, (-0.5 * Math.Pow(((X[i - 1] - Media) / Desviacion), 2)));
511.                     Resultado[i - 1] = (int)Math.Round(f * X[i - 1]);
512.                 }
513.
514.                 return Resultado; //Se devuelve el vector con todos los valores
515.             }
516.
517.             int[] DistribucionEnRampa(double Media, int tiempo, bool Sentido)
518.             {
519.                 //Función que obtiene la distribución en Rampa según la media, y
    el sentido de la pendiente
520.                 int[] Resultado = new int[tiempo + 1];
521.                 double[] X = new Double[tiempo + 1];
522.
523.                 double f;
524.                 for (int i = 1; i <= (tiempo + 1); i++)
525.                 {
526.                     //Se calcula según si es de Pendiente Negativa o Positiva.
527.                     if (Sentido) f = 2 * (Media / tiempo) * (i - 1); //Ascendent
    e
528.                     else f = (2 * Media / tiempo) * (tiempo - (i - 1));        //
    Descendente
529.                     Resultado[i - 1] = (int)Math.Round(f);
530.                 }
531.
532.                 return Resultado;//Se devuelve el vector con todos los valores
533.             }
534.             int[] DistribucionConstante(double Media, int tiempo)
535.             {
```

```
536.                 //Función que obtiene la distribución Constante según la media
537.                 int[] Resultado = new int[tiempo + 1];
538.                 for (int i = 1; i <= (tiempo + 1); i++)
539.                 {
540.                     Resultado[i - 1] = (int)Media; //Todos los valores son
    iguales
541.                 }
542.                 return Resultado;//Se devuelve el vector con todos los valores
543.             }
544.
545.
546.             int[] Inicializar(int[] Celda)
547.             {//Recibe un vector cuyos valores se ponen a 0.
548.                 for(int i = 0; i < Celda.Length; i++)
549.                 {
550.                     Celda[i] = 0;
551.                 }
552.                 return Celda;
553.             }
554.             string CrearString(int[] Celda)
555.             {//Crea una cadena separada por comas a partir de un vector de
    enteros
556.                 string Res="";
557.                 for(int i= 0; i < (Celda.Length-1); i++)
558.                 {
559.                     Res += Celda[i].ToString() + ",";
560.                 }
561.                 Res += Celda.Last().ToString();
562.                 return Res;
563.             }
564.             int ObtencionEntradas(int Tramo, int i, double[] Importancia)
565.             {
566.                 int Salida = 0;
567.                 Salida = (int)Math.Round((Importancia[i]/100)*UltimosValores[i])
    ;
568.                 //if(Salida>0)MessageBox.Show(Salida.ToString());
569.                 UltimosValores[i] -= Salida;
570.
571.                 return Salida;
572.             }
573.         }
574.     }
575.
```

Anexo III: Código Diseñador

```
1. namespace ProjecteTFG
2. {
3.     //En este archivo se observa la disposición inicial de todos los elementos.
4.     //Su contenido es necesario para conocer ante todo las propiedades de los
   elementos.
5.     partial class Form1
6.     {
7.         /// <summary>
8.         /// Variable del diseñador necesaria.
9.         /// </summary>
10.             private System.ComponentModel.IContainer components = null;
11.
12.             /// <summary>
13.             /// Limpiar los recursos que se estén usando.
14.             /// </summary>
15.             /// <param name="disposing">true si los recursos administrados se
   deben desechar; false en caso contrario.</param>
16.             protected override void Dispose(bool disposing)
17.             {
18.                 if (disposing && (components != null))
19.                 {
20.                     components.Dispose();
21.                 }
22.                 base.Dispose(disposing);
23.             }
24.
25.             #region Código generado por el Diseñador de Windows Forms
26.
27.             /// <summary>
28.             /// Método necesario para admitir el Diseñador. No se puede
   modificar
29.             /// el contenido de este método con el editor de código.
30.             /// </summary>
31.             private void InitializeComponent()
32.             {
33.                 this.components = new System.ComponentModel.Container();
34.                 System.Windows.Forms.DataVisualization.Charting.ChartArea chartA
   rea1 = new System.Windows.Forms.DataVisualization.Charting.ChartArea();
35.                 System.Windows.Forms.DataVisualization.Charting.Legend legend1 =
    new System.Windows.Forms.DataVisualization.Charting.Legend();
36.                 System.Windows.Forms.DataVisualization.Charting.Series series1 =
    new System.Windows.Forms.DataVisualization.Charting.Series();
37.                 System.ComponentModel.ComponentResourceManager resources = new S
   ystem.ComponentModel.ComponentResourceManager(typeof(Form1));
38.                 this.gMapControl1 = new GMap.NET.WindowsForms.GMapControl();
39.                 this.txtdescripcion = new System.Windows.Forms.TextBox();
40.                 this.label1 = new System.Windows.Forms.Label();
41.                 this.txtlatitud = new System.Windows.Forms.TextBox();
42.                 this.label2 = new System.Windows.Forms.Label();
43.                 this.label3 = new System.Windows.Forms.Label();
44.                 this.txtlongitud = new System.Windows.Forms.TextBox();
45.                 this.dataGridView1 = new System.Windows.Forms.DataGridView();
46.                 this.timer1 = new System.Windows.Forms.Timer(this.components);
47.                 this.panel2 = new System.Windows.Forms.Panel();
48.                 this.txtVelMed = new System.Windows.Forms.TextBox();
49.                 this.numerico = new System.Windows.Forms.NumericUpDown();
```

```
50.                 this.tipoOut = new System.Windows.Forms.ListBox();
51.                 this.label4 = new System.Windows.Forms.Label();
52.                 this.label11 = new System.Windows.Forms.Label();
53.                 this.btnAgregar = new System.Windows.Forms.Button();
54.                 this.label6 = new System.Windows.Forms.Label();
55.                 this.btnOk = new System.Windows.Forms.Button();
56.                 this.textBox1 = new System.Windows.Forms.TextBox();
57.                 this.LNewSim = new System.Windows.Forms.Label();
58.                 this.panel1 = new System.Windows.Forms.Panel();
59.                 this.btnLoadSim = new System.Windows.Forms.Button();
60.                 this.btnNewSim = new System.Windows.Forms.Button();
61.                 this.LInstruccion1 = new System.Windows.Forms.Label();
62.                 this.LInstruccion2 = new System.Windows.Forms.Label();
63.                 this.LNombreProyecto = new System.Windows.Forms.Label();
64.                 this.LCuenta = new System.Windows.Forms.Label();
65.                 this.listaEntradas = new System.Windows.Forms.ListBox();
66.                 this.label8 = new System.Windows.Forms.Label();
67.                 this.label9 = new System.Windows.Forms.Label();
68.                 this.listaSalidas = new System.Windows.Forms.ListBox();
69.                 this.listaIntermedios = new System.Windows.Forms.ListBox();
70.                 this.label10 = new System.Windows.Forms.Label();
71.                 this.comboDis = new System.Windows.Forms.ComboBox();
72.                 this.Graf = new System.Windows.Forms.DataVisualization.Charting.
    Chart();
73.                 this.label12 = new System.Windows.Forms.Label();
74.                 this.numerico2 = new System.Windows.Forms.NumericUpDown();
75.                 this.numerico3 = new System.Windows.Forms.NumericUpDown();
76.                 this.label13 = new System.Windows.Forms.Label();
77.                 this.ListLog = new System.Windows.Forms.GroupBox();
78.                 this.Asc = new System.Windows.Forms.RadioButton();
79.                 this.Desc = new System.Windows.Forms.RadioButton();
80.                 this.Temporizador = new System.Windows.Forms.Timer(this.componen
    ts);
81.                 this.folderBrowserDialog1 = new System.Windows.Forms.FolderBrows
    erDialog();
82.                 this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog()
    ;
83.                 this.saveFileDialog1 = new System.Windows.Forms.SaveFileDialog()
    ;
84.                 this.trackBar1 = new System.Windows.Forms.TrackBar();
85.                 this.bFinalizar = new System.Windows.Forms.Button();
86.                 this.bDetener = new System.Windows.Forms.Button();
87.                 this.bPlay = new System.Windows.Forms.Button();
88.                 this.btnCambio = new System.Windows.Forms.Button();
89.                 this.label5 = new System.Windows.Forms.Label();
90.                 ((System.ComponentModel.ISupportInitialize)(this.dataGridView1))
    .BeginInit();
91.                 this.panel2.SuspendLayout();
92.                 ((System.ComponentModel.ISupportInitialize)(this.numerico)).Begi
    nInit();
93.                 this.panel1.SuspendLayout();
94.                 ((System.ComponentModel.ISupportInitialize)(this.Graf)).BeginIni
    t();
95.                 ((System.ComponentModel.ISupportInitialize)(this.numerico2)).Beg
    inInit();
96.                 ((System.ComponentModel.ISupportInitialize)(this.numerico3)).Beg
    inInit();
97.                 this.ListLog.SuspendLayout();
98.                 ((System.ComponentModel.ISupportInitialize)(this.trackBar1)).Beg
    inInit();
99.                 this.SuspendLayout();
```

```
100.                //
101.                // gMapControl1
102.                //
103.                this.gMapControl1.Bearing = 0F;
104.                this.gMapControl1.CanDragMap = true;
105.                this.gMapControl1.EmptyTileColor = System.Drawing.Color.Navy;
106.                this.gMapControl1.GrayScaleMode = false;
107.                this.gMapControl1.HelperLineOption = GMap.NET.WindowsForms.Helpe
    rLineOptions.DontShow;
108.                this.gMapControl1.LevelsKeepInMemmory = 5;
109.                this.gMapControl1.Location = new System.Drawing.Point(241, 179);
110.                this.gMapControl1.Margin = new System.Windows.Forms.Padding(2, 2
    , 2, 2);
111.                this.gMapControl1.MarkersEnabled = true;
112.                this.gMapControl1.MaxZoom = 2;
113.                this.gMapControl1.MinZoom = 2;
114.                this.gMapControl1.MouseWheelZoomEnabled = true;
115.                this.gMapControl1.MouseWheelZoomType = GMap.NET.MouseWheelZoomTy
    pe.MousePositionAndCenter;
116.                this.gMapControl1.Name = "gMapControl1";
117.                this.gMapControl1.NegativeMode = false;
118.                this.gMapControl1.PolygonsEnabled = true;
119.                this.gMapControl1.RetryLoadTile = 0;
120.                this.gMapControl1.RoutesEnabled = true;
121.                this.gMapControl1.ScaleMode = GMap.NET.WindowsForms.ScaleModes.I
    nteger;
122.                this.gMapControl1.SelectedAreaFillColor = System.Drawing.Color.F
    romArgb(((int)(((byte)(33)))), ((int)(((byte)(65)))), ((int)(((byte)(105)))), ((i
    nt)(((byte)(225)))));
123.                this.gMapControl1.ShowTileGridLines = false;
124.                this.gMapControl1.Size = new System.Drawing.Size(721, 361);
125.                this.gMapControl1.TabIndex = 0;
126.                this.gMapControl1.Visible = false;
127.                this.gMapControl1.Zoom = 0D;
128.                this.gMapControl1.Load += new System.EventHandler(this.gMapContr
    ol1_Load);
129.                this.gMapControl1.MouseDoubleClick += new System.Windows.Forms.M
    ouseEventHandler(this.gMapControl1_MouseDoubleClick);
130.                //
131.                // txtdescripcion
132.                //
133.                this.txtdescripcion.Font = new System.Drawing.Font("Franklin
    Gothic Medium", 12F);
134.                this.txtdescripcion.Location = new System.Drawing.Point(13, 170)
    ;
135.                this.txtdescripcion.Margin = new System.Windows.Forms.Padding(2,
     2, 2, 2);
136.                this.txtdescripcion.Name = "txtdescripcion";
137.                this.txtdescripcion.Size = new System.Drawing.Size(187, 26);
138.                this.txtdescripcion.TabIndex = 3;
139.                this.txtdescripcion.Visible = false;
140.                this.txtdescripcion.TextChanged += new System.EventHandler(this.
    textBox1_TextChanged);
141.                //
142.                // label1
143.                //
144.                this.label1.AutoSize = true;
145.                this.label1.Font = new System.Drawing.Font("Franklin Gothic
    Medium",
    12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
```

```
146.                 this.label1.ForeColor = System.Drawing.Color.White;
147.                 this.label1.Location = new System.Drawing.Point(20, 148);
148.                 this.label1.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
149.                 this.label1.Name = "label1";
150.                 this.label1.Size = new System.Drawing.Size(88, 21);
151.                 this.label1.TabIndex = 4;
152.                 this.label1.Text = "Descripción";
153.                 this.label1.Visible = false;
154.                 //
155.                 // txtlatitud
156.                 //
157.                 this.txtlatitud.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
158.                 this.txtlatitud.Location = new System.Drawing.Point(13, 216);
159.                 this.txtlatitud.Margin = new System.Windows.Forms.Padding(2, 2,
    2, 2);
160.                 this.txtlatitud.Name = "txtlatitud";
161.                 this.txtlatitud.Size = new System.Drawing.Size(187, 26);
162.                 this.txtlatitud.TabIndex = 5;
163.                 this.txtlatitud.Visible = false;
164.                 this.txtlatitud.KeyPress += new System.Windows.Forms.KeyPressEve
    ntHandler(this.txtlatitud_KeyPress);
165.                 //
166.                 // label2
167.                 //
168.                 this.label2.AutoSize = true;
169.                 this.label2.Font = new System.Drawing.Font("Franklin Gothic
    Medium",
    12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
170.                 this.label2.ForeColor = System.Drawing.Color.White;
171.                 this.label2.Location = new System.Drawing.Point(20, 194);
172.                 this.label2.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
173.                 this.label2.Name = "label2";
174.                 this.label2.Size = new System.Drawing.Size(59, 21);
175.                 this.label2.TabIndex = 6;
176.                 this.label2.Text = "Latitud";
177.                 this.label2.Visible = false;
178.                 //
179.                 // label3
180.                 //
181.                 this.label3.AutoSize = true;
182.                 this.label3.Font = new System.Drawing.Font("Franklin Gothic
    Medium",
    12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
183.                 this.label3.ForeColor = System.Drawing.Color.White;
184.                 this.label3.Location = new System.Drawing.Point(20, 243);
185.                 this.label3.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
186.                 this.label3.Name = "label3";
187.                 this.label3.Size = new System.Drawing.Size(70, 21);
188.                 this.label3.TabIndex = 8;
189.                 this.label3.Text = "Longitud";
190.                 this.label3.Visible = false;
191.                 //
192.                 // txtlongitud
193.                 //
```

```
194.                 this.txtlongitud.Font = new System.Drawing.Font("Franklin Gothic
   Medium", 12F);
195.                 this.txtlongitud.Location = new System.Drawing.Point(13, 265);
196.                 this.txtlongitud.Margin = new System.Windows.Forms.Padding(2, 2,
    2, 2);
197.                 this.txtlongitud.Name = "txtlongitud";
198.                 this.txtlongitud.Size = new System.Drawing.Size(187, 26);
199.                 this.txtlongitud.TabIndex = 7;
200.                 this.txtlongitud.Visible = false;
201.                 this.txtlongitud.KeyPress += new System.Windows.Forms.KeyPressEv
   entHandler(this.txtlongitud_KeyPress);
202.                 //
203.                 // dataGridView1
204.                 //
205.                 this.dataGridView1.AllowUserToAddRows = false;
206.                 this.dataGridView1.AutoSizeColumnsMode = System.Windows.Forms.Da
   taGridViewAutoSizeColumnsMode.Fill;
207.                 this.dataGridView1.ColumnHeadersHeightSizeMode = System.Windows.
   Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
208.                 this.dataGridView1.Location = new System.Drawing.Point(241, 179)
   ;
209.                 this.dataGridView1.Margin = new System.Windows.Forms.Padding(2,
   2, 2, 2);
210.                 this.dataGridView1.Name = "dataGridView1";
211.                 this.dataGridView1.ReadOnly = true;
212.                 this.dataGridView1.RowTemplate.Height = 28;
213.                 this.dataGridView1.Size = new System.Drawing.Size(721, 361);
214.                 this.dataGridView1.TabIndex = 9;
215.                 this.dataGridView1.Visible = false;
216.                 this.dataGridView1.CellMouseClick += new System.Windows.Forms.Da
   taGridViewCellMouseEventHandler(this.SeleccionarRegistro);
217.                 //
218.                 // timer1
219.                 //
220.                 this.timer1.Enabled = true;
221.                 this.timer1.Interval = 50;
222.                 this.timer1.Tick += new System.EventHandler(this.timer1_Tick);
223.                 //
224.                 // panel2
225.                 //
226.                 this.panel2.BackColor = System.Drawing.Color.FromArgb(((int)(((b
   yte)(26)))), ((int)(((byte)(32)))), ((int)(((byte)(40)))));
227.                 this.panel2.Controls.Add(this.txtVelMed);
228.                 this.panel2.Controls.Add(this.numerico);
229.                 this.panel2.Controls.Add(this.tipoOut);
230.                 this.panel2.Controls.Add(this.label4);
231.                 this.panel2.Controls.Add(this.label11);
232.                 this.panel2.Controls.Add(this.btnAgregar);
233.                 this.panel2.Controls.Add(this.label6);
234.                 this.panel2.Controls.Add(this.btnOk);
235.                 this.panel2.Controls.Add(this.textBox1);
236.                 this.panel2.Controls.Add(this.LNewSim);
237.                 this.panel2.Controls.Add(this.txtdescripcion);
238.                 this.panel2.Controls.Add(this.label1);
239.                 this.panel2.Controls.Add(this.txtlatitud);
240.                 this.panel2.Controls.Add(this.label2);
241.                 this.panel2.Controls.Add(this.txtlongitud);
242.                 this.panel2.Controls.Add(this.label3);
243.                 this.panel2.Dock = System.Windows.Forms.DockStyle.Left;
244.                 this.panel2.ForeColor = System.Drawing.Color.White;
245.                 this.panel2.Location = new System.Drawing.Point(0, 0);
```

```
246.                 this.panel2.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2
   );
247.                 this.panel2.Name = "panel2";
248.                 this.panel2.Size = new System.Drawing.Size(208, 615);
249.                 this.panel2.TabIndex = 19;
250.                 //
251.                 // txtVelMed
252.                 //
253.                 this.txtVelMed.Font = new System.Drawing.Font("Franklin Gothic
   Medium", 12F);
254.                 this.txtVelMed.Location = new System.Drawing.Point(11, 502);
255.                 this.txtVelMed.Margin = new System.Windows.Forms.Padding(2, 2, 2
   , 2);
256.                 this.txtVelMed.Name = "txtVelMed";
257.                 this.txtVelMed.Size = new System.Drawing.Size(187, 26);
258.                 this.txtVelMed.TabIndex = 34;
259.                 this.txtVelMed.Visible = false;
260.                 this.txtVelMed.KeyPress += new System.Windows.Forms.KeyPressEven
   tHandler(this.txtVelMed_KeyPress);
261.                 //
262.                 // numerico
263.                 //
264.                 this.numerico.Font = new System.Drawing.Font("Franklin Gothic
   Medium", 12F);
265.                 this.numerico.Location = new System.Drawing.Point(13, 317);
266.                 this.numerico.Margin = new System.Windows.Forms.Padding(2, 2, 2,
    2);
267.                 this.numerico.Name = "numerico";
268.                 this.numerico.Size = new System.Drawing.Size(80, 26);
269.                 this.numerico.TabIndex = 28;
270.                 this.numerico.Visible = false;
271.                 this.numerico.ValueChanged += new System.EventHandler(this.numer
   ico_ValueChanged);
272.                 //
273.                 // tipoOut
274.                 //
275.                 this.tipoOut.Font = new System.Drawing.Font("Franklin Gothic
   Medium", 12F);
276.                 this.tipoOut.FormattingEnabled = true;
277.                 this.tipoOut.ItemHeight = 21;
278.                 this.tipoOut.Location = new System.Drawing.Point(13, 372);
279.                 this.tipoOut.Margin = new System.Windows.Forms.Padding(2, 2, 2,
   2);
280.                 this.tipoOut.Name = "tipoOut";
281.                 this.tipoOut.Size = new System.Drawing.Size(187, 88);
282.                 this.tipoOut.TabIndex = 33;
283.                 this.tipoOut.Visible = false;
284.                 //
285.                 // label4
286.                 //
287.                 this.label4.AutoSize = true;
288.                 this.label4.Font = new System.Drawing.Font("Franklin Gothic
   Medium",
   12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
   (0)));
289.                 this.label4.ForeColor = System.Drawing.Color.White;
290.                 this.label4.Location = new System.Drawing.Point(18, 481);
291.                 this.label4.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
   );
292.                 this.label4.Name = "label4";
293.                 this.label4.Size = new System.Drawing.Size(179, 21);
```

```
294.                 this.label4.TabIndex = 35;
295.                 this.label4.Text = "Velocidad Media (Km/h)";
296.                 this.label4.Visible = false;
297.                 //
298.                 // label11
299.                 //
300.                 this.label11.AutoSize = true;
301.                 this.label11.Font = new System.Drawing.Font("Franklin Gothic
    Medium",
    12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
302.                 this.label11.ForeColor = System.Drawing.Color.White;
303.                 this.label11.Location = new System.Drawing.Point(20, 350);
304.                 this.label11.Margin = new System.Windows.Forms.Padding(2, 0, 2,
    0);
305.                 this.label11.Name = "label11";
306.                 this.label11.Size = new System.Drawing.Size(109, 21);
307.                 this.label11.TabIndex = 32;
308.                 this.label11.Text = "Tipo Semáforo";
309.                 this.label11.Visible = false;
310.                 //
311.                 // btnAgregar
312.                 //
313.                 this.btnAgregar.Enabled = false;
314.                 this.btnAgregar.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
315.                 this.btnAgregar.ForeColor = System.Drawing.Color.FromArgb(((int)
    (((byte)(26)))), ((int)(((byte)(32)))), ((int)(((byte)(40)))));
316.                 this.btnAgregar.Image = global::ProjecteTFG.Properties.Resources
    .agregar;
317.                 this.btnAgregar.Location = new System.Drawing.Point(68, 545);
318.                 this.btnAgregar.Margin = new System.Windows.Forms.Padding(2, 2,
    2, 2);
319.                 this.btnAgregar.Name = "btnAgregar";
320.                 this.btnAgregar.Size = new System.Drawing.Size(52, 47);
321.                 this.btnAgregar.TabIndex = 1;
322.                 this.btnAgregar.UseVisualStyleBackColor = true;
323.                 this.btnAgregar.Visible = false;
324.                 this.btnAgregar.Click += new System.EventHandler(this.btnAgregar
    _Click);
325.                 //
326.                 // label6
327.                 //
328.                 this.label6.AutoSize = true;
329.                 this.label6.Font = new System.Drawing.Font("Franklin Gothic
    Medium",
    12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
330.                 this.label6.ForeColor = System.Drawing.Color.White;
331.                 this.label6.Location = new System.Drawing.Point(20, 295);
332.                 this.label6.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
333.                 this.label6.Name = "label6";
334.                 this.label6.Size = new System.Drawing.Size(140, 21);
335.                 this.label6.TabIndex = 29;
336.                 this.label6.Text = "Número de Carriles";
337.                 this.label6.Visible = false;
338.                 //
339.                 // btnOk
340.                 //
341.                 this.btnOk.Enabled = false;
342.                 this.btnOk.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
```

```
343.                 this.btnOk.ForeColor = System.Drawing.Color.FromArgb(((int)(((by
    te)(26)))), ((int)(((byte)(32)))), ((int)(((byte)(40)))));
344.                 this.btnOk.Image = global::ProjecteTFG.Properties.Resources.sigu
    iente2;
345.                 this.btnOk.Location = new System.Drawing.Point(21, 79);
346.                 this.btnOk.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2)
    ;
347.                 this.btnOk.Name = "btnOk";
348.                 this.btnOk.Size = new System.Drawing.Size(56, 55);
349.                 this.btnOk.TabIndex = 2;
350.                 this.btnOk.UseVisualStyleBackColor = true;
351.                 this.btnOk.Visible = false;
352.                 this.btnOk.Click += new System.EventHandler(this.btnOk_Click);
353.                 //
354.                 // textBox1
355.                 //
356.                 this.textBox1.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
357.                 this.textBox1.Location = new System.Drawing.Point(13, 44);
358.                 this.textBox1.Margin = new System.Windows.Forms.Padding(2, 2, 2,
     2);
359.                 this.textBox1.Name = "textBox1";
360.                 this.textBox1.Size = new System.Drawing.Size(187, 26);
361.                 this.textBox1.TabIndex = 1;
362.                 this.textBox1.Visible = false;
363.                 this.textBox1.TextChanged += new System.EventHandler(this.textBo
    x1_TextChanged_1);
364.                 //
365.                 // LNewSim
366.                 //
367.                 this.LNewSim.AutoSize = true;
368.                 this.LNewSim.Font = new System.Drawing.Font("Franklin Gothic
    Medium",
    12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
369.                 this.LNewSim.Location = new System.Drawing.Point(19, 22);
370.                 this.LNewSim.Margin = new System.Windows.Forms.Padding(2, 0, 2,
    0);
371.                 this.LNewSim.Name = "LNewSim";
372.                 this.LNewSim.Size = new System.Drawing.Size(64, 21);
373.                 this.LNewSim.TabIndex = 0;
374.                 this.LNewSim.Text = "Nombre";
375.                 this.LNewSim.Visible = false;
376.                 //
377.                 // panel1
378.                 //
379.                 this.panel1.BackColor = System.Drawing.Color.FromArgb(((int)(((b
    yte)(0)))), ((int)(((byte)(80)))), ((int)(((byte)(200)))));
380.                 this.panel1.Controls.Add(this.btnLoadSim);
381.                 this.panel1.Controls.Add(this.btnNewSim);
382.                 this.panel1.Dock = System.Windows.Forms.DockStyle.Top;
383.                 this.panel1.Location = new System.Drawing.Point(208, 0);
384.                 this.panel1.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2
    );
385.                 this.panel1.Name = "panel1";
386.                 this.panel1.Size = new System.Drawing.Size(863, 79);
387.                 this.panel1.TabIndex = 20;
388.                 //
389.                 // btnLoadSim
390.                 //
```

```
391.                 this.btnLoadSim.BackColor = System.Drawing.Color.FromArgb(((int)
    (((byte)(0)))), ((int)(((byte)(80)))), ((int)(((byte)(200)))));
392.                 this.btnLoadSim.BackgroundImageLayout = System.Windows.Forms.Ima
    geLayout.None;
393.                 this.btnLoadSim.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
394.                 this.btnLoadSim.ForeColor = System.Drawing.Color.FromArgb(((int)
    (((byte)(0)))), ((int)(((byte)(80)))), ((int)(((byte)(200)))));
395.                 this.btnLoadSim.Image = global::ProjecteTFG.Properties.Resources
    .filesystems_thefolder_636;
396.                 this.btnLoadSim.Location = new System.Drawing.Point(98, 6);
397.                 this.btnLoadSim.Margin = new System.Windows.Forms.Padding(2, 2,
    2, 2);
398.                 this.btnLoadSim.Name = "btnLoadSim";
399.                 this.btnLoadSim.Size = new System.Drawing.Size(65, 58);
400.                 this.btnLoadSim.TabIndex = 1;
401.                 this.btnLoadSim.UseVisualStyleBackColor = false;
402.                 this.btnLoadSim.Click += new System.EventHandler(this.btnLoadSim
    _Click);
403.                 //
404.                 // btnNewSim
405.                 //
406.                 this.btnNewSim.BackColor = System.Drawing.Color.FromArgb(((int)(
    ((byte)(0)))), ((int)(((byte)(80)))), ((int)(((byte)(200)))));
407.                 this.btnNewSim.BackgroundImageLayout = System.Windows.Forms.Imag
    eLayout.None;
408.                 this.btnNewSim.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
409.                 this.btnNewSim.ForeColor = System.Drawing.Color.FromArgb(((int)(
    ((byte)(0)))), ((int)(((byte)(80)))), ((int)(((byte)(200)))));
410.                 this.btnNewSim.Image = global::ProjecteTFG.Properties.Resources.
    NuevoArchivo;
411.                 this.btnNewSim.Location = new System.Drawing.Point(4, 3);
412.                 this.btnNewSim.Margin = new System.Windows.Forms.Padding(2, 2, 2
    , 2);
413.                 this.btnNewSim.Name = "btnNewSim";
414.                 this.btnNewSim.Size = new System.Drawing.Size(73, 62);
415.                 this.btnNewSim.TabIndex = 0;
416.                 this.btnNewSim.UseVisualStyleBackColor = false;
417.                 this.btnNewSim.Click += new System.EventHandler(this.btnNewSim_C
    lick);
418.                 //
419.                 // LInstruccion1
420.                 //
421.                 this.LInstruccion1.AutoSize = true;
422.                 this.LInstruccion1.Font = new System.Drawing.Font("Franklin
    Gothic Medium", 12F, System.Drawing.FontStyle.Bold);
423.                 this.LInstruccion1.ForeColor = System.Drawing.Color.White;
424.                 this.LInstruccion1.Location = new System.Drawing.Point(245, 114)
    ;
425.                 this.LInstruccion1.Margin = new System.Windows.Forms.Padding(2,
    0, 2, 0);
426.                 this.LInstruccion1.Name = "LInstruccion1";
427.                 this.LInstruccion1.Size = new System.Drawing.Size(0, 21);
428.                 this.LInstruccion1.TabIndex = 22;
429.                 this.LInstruccion1.Visible = false;
430.                 //
431.                 // LInstruccion2
432.                 //
433.                 this.LInstruccion2.AutoSize = true;
434.                 this.LInstruccion2.Font = new System.Drawing.Font("Franklin
    Gothic Medium", 12F);
435.                 this.LInstruccion2.ForeColor = System.Drawing.Color.White;
```

```
436.                 this.LInstruccion2.Location = new System.Drawing.Point(245, 144)
    ;
437.                 this.LInstruccion2.Margin = new System.Windows.Forms.Padding(2,
    0, 2, 0);
438.                 this.LInstruccion2.Name = "LInstruccion2";
439.                 this.LInstruccion2.Size = new System.Drawing.Size(106, 21);
440.                 this.LInstruccion2.TabIndex = 23;
441.                 this.LInstruccion2.Text = "N° de Tramos:";
442.                 this.LInstruccion2.Visible = false;
443.                 //
444.                 // LNombreProyecto
445.                 //
446.                 this.LNombreProyecto.AutoSize = true;
447.                 this.LNombreProyecto.Font = new System.Drawing.Font("Franklin
    Gothic Medium",
    14F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)
    ));
448.                 this.LNombreProyecto.ForeColor = System.Drawing.Color.White;
449.                 this.LNombreProyecto.Location = new System.Drawing.Point(245, 91
    );
450.                 this.LNombreProyecto.Margin = new System.Windows.Forms.Padding(2
    , 0, 2, 0);
451.                 this.LNombreProyecto.Name = "LNombreProyecto";
452.                 this.LNombreProyecto.Size = new System.Drawing.Size(0, 24);
453.                 this.LNombreProyecto.TabIndex = 24;
454.                 //
455.                 // LCuenta
456.                 //
457.                 this.LCuenta.AutoSize = true;
458.                 this.LCuenta.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
459.                 this.LCuenta.ForeColor = System.Drawing.Color.White;
460.                 this.LCuenta.Location = new System.Drawing.Point(354, 144);
461.                 this.LCuenta.Margin = new System.Windows.Forms.Padding(2, 0, 2,
    0);
462.                 this.LCuenta.Name = "LCuenta";
463.                 this.LCuenta.Size = new System.Drawing.Size(19, 21);
464.                 this.LCuenta.TabIndex = 25;
465.                 this.LCuenta.Text = "0";
466.                 this.LCuenta.Visible = false;
467.                 //
468.                 // listaEntradas
469.                 //
470.                 this.listaEntradas.FormattingEnabled = true;
471.                 this.listaEntradas.Location = new System.Drawing.Point(305, 329)
    ;
472.                 this.listaEntradas.Margin = new System.Windows.Forms.Padding(2,
    2, 2, 2);
473.                 this.listaEntradas.Name = "listaEntradas";
474.                 this.listaEntradas.SelectionMode = System.Windows.Forms.Selectio
    nMode.MultiSimple;
475.                 this.listaEntradas.Size = new System.Drawing.Size(186, 56);
476.                 this.listaEntradas.TabIndex = 29;
477.                 this.listaEntradas.Visible = false;
478.                 //
479.                 // label8
480.                 //
481.                 this.label8.AutoSize = true;
482.                 this.label8.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
483.                 this.label8.ForeColor = System.Drawing.Color.White;
```

```
484.                 this.label8.Location = new System.Drawing.Point(318, 311);
485.                 this.label8.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
486.                 this.label8.Name = "label8";
487.                 this.label8.Size = new System.Drawing.Size(65, 21);
488.                 this.label8.TabIndex = 30;
489.                 this.label8.Text = "Entrada";
490.                 this.label8.Visible = false;
491.                 //
492.                 // label9
493.                 //
494.                 this.label9.AutoSize = true;
495.                 this.label9.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
496.                 this.label9.ForeColor = System.Drawing.Color.White;
497.                 this.label9.Location = new System.Drawing.Point(245, 314);
498.                 this.label9.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
499.                 this.label9.Name = "label9";
500.                 this.label9.Size = new System.Drawing.Size(61, 21);
501.                 this.label9.TabIndex = 31;
502.                 this.label9.Text = "Salidas";
503.                 this.label9.Visible = false;
504.                 //
505.                 // listaSalidas
506.                 //
507.                 this.listaSalidas.Font = new System.Drawing.Font("Franklin
    Gothic Medium", 12F);
508.                 this.listaSalidas.FormattingEnabled = true;
509.                 this.listaSalidas.ItemHeight = 21;
510.                 this.listaSalidas.Location = new System.Drawing.Point(248, 342);
511.                 this.listaSalidas.Margin = new System.Windows.Forms.Padding(2, 2
    , 2, 2);
512.                 this.listaSalidas.Name = "listaSalidas";
513.                 this.listaSalidas.Size = new System.Drawing.Size(186, 46);
514.                 this.listaSalidas.TabIndex = 32;
515.                 this.listaSalidas.Visible = false;
516.                 this.listaSalidas.SelectedIndexChanged += new System.EventHandle
    r(this.listaSalidas_SelectedIndexChanged);
517.                 //
518.                 // listaIntermedios
519.                 //
520.                 this.listaIntermedios.Font = new System.Drawing.Font("Franklin
    Gothic Medium", 12F);
521.                 this.listaIntermedios.FormattingEnabled = true;
522.                 this.listaIntermedios.ItemHeight = 21;
523.                 this.listaIntermedios.Location = new System.Drawing.Point(519, 3
    29);
524.                 this.listaIntermedios.Margin = new System.Windows.Forms.Padding(
    2, 2, 2, 2);
525.                 this.listaIntermedios.Name = "listaIntermedios";
526.                 this.listaIntermedios.Size = new System.Drawing.Size(186, 46);
527.                 this.listaIntermedios.TabIndex = 33;
528.                 this.listaIntermedios.Visible = false;
529.                 //
530.                 // label10
531.                 //
532.                 this.label10.AutoSize = true;
533.                 this.label10.BackColor = System.Drawing.Color.White;
534.                 this.label10.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
```

```
535.                 this.label10.ForeColor = System.Drawing.Color.White;
536.                 this.label10.Location = new System.Drawing.Point(583, 311);
537.                 this.label10.Margin = new System.Windows.Forms.Padding(2, 0, 2,
    0);
538.                 this.label10.Name = "label10";
539.                 this.label10.Size = new System.Drawing.Size(84, 21);
540.                 this.label10.TabIndex = 34;
541.                 this.label10.Text = "Intermedio";
542.                 this.label10.Visible = false;
543.                 //
544.                 // comboDis
545.                 //
546.                 this.comboDis.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
547.                 this.comboDis.FormattingEnabled = true;
548.                 this.comboDis.Location = new System.Drawing.Point(241, 192);
549.                 this.comboDis.Margin = new System.Windows.Forms.Padding(2, 2, 2,
     2);
550.                 this.comboDis.Name = "comboDis";
551.                 this.comboDis.Size = new System.Drawing.Size(247, 29);
552.                 this.comboDis.TabIndex = 35;
553.                 this.comboDis.Visible = false;
554.                 this.comboDis.SelectedIndexChanged += new System.EventHandler(th
    is.comboDis_SelectedIndexChanged);
555.                 //
556.                 // Graf
557.                 //
558.                 this.Graf.BackGradientStyle = System.Windows.Forms.DataVisualiza
    tion.Charting.GradientStyle.LeftRight;
559.                 this.Graf.BorderSkin.BorderColor = System.Drawing.Color.Maroon;
560.                 chartArea1.Name = "ChartArea1";
561.                 this.Graf.ChartAreas.Add(chartArea1);
562.                 legend1.Name = "Legend1";
563.                 this.Graf.Legends.Add(legend1);
564.                 this.Graf.Location = new System.Drawing.Point(241, 233);
565.                 this.Graf.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
566.                 this.Graf.Name = "Graf";
567.                 series1.ChartArea = "ChartArea1";
568.                 series1.ChartType = System.Windows.Forms.DataVisualization.Chart
    ing.SeriesChartType.Line;
569.                 series1.IsXValueIndexed = true;
570.                 series1.Legend = "Legend1";
571.                 series1.Name = "Vehículos";
572.                 this.Graf.Series.Add(series1);
573.                 this.Graf.Size = new System.Drawing.Size(502, 238);
574.                 this.Graf.TabIndex = 36;
575.                 this.Graf.Text = "chart1";
576.                 this.Graf.Visible = false;
577.                 //
578.                 // label12
579.                 //
580.                 this.label12.AutoSize = true;
581.                 this.label12.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
582.                 this.label12.ForeColor = System.Drawing.Color.White;
583.                 this.label12.Location = new System.Drawing.Point(779, 314);
584.                 this.label12.Margin = new System.Windows.Forms.Padding(2, 0, 2,
    0);
585.                 this.label12.Name = "label12";
586.                 this.label12.Size = new System.Drawing.Size(140, 21);
587.                 this.label12.TabIndex = 37;
```

```
588.                 this.label12.Text = "Numero de Carriles";
589.                 this.label12.Visible = false;
590.                 //
591.                 // numerico2
592.                 //
593.                 this.numerico2.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
594.                 this.numerico2.Location = new System.Drawing.Point(781, 337);
595.                 this.numerico2.Margin = new System.Windows.Forms.Padding(2, 2, 2
    , 2);
596.                 this.numerico2.Name = "numerico2";
597.                 this.numerico2.Size = new System.Drawing.Size(80, 26);
598.                 this.numerico2.TabIndex = 38;
599.                 this.numerico2.Visible = false;
600.                 //
601.                 // numerico3
602.                 //
603.                 this.numerico3.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
604.                 this.numerico3.Location = new System.Drawing.Point(781, 242);
605.                 this.numerico3.Margin = new System.Windows.Forms.Padding(2, 2, 2
    , 2);
606.                 this.numerico3.Name = "numerico3";
607.                 this.numerico3.Size = new System.Drawing.Size(80, 26);
608.                 this.numerico3.TabIndex = 34;
609.                 this.numerico3.Visible = false;
610.                 //
611.                 // label13
612.                 //
613.                 this.label13.AutoSize = true;
614.                 this.label13.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
615.                 this.label13.ForeColor = System.Drawing.Color.White;
616.                 this.label13.Location = new System.Drawing.Point(779, 218);
617.                 this.label13.Margin = new System.Windows.Forms.Padding(2, 0, 2,
    0);
618.                 this.label13.Name = "label13";
619.                 this.label13.Size = new System.Drawing.Size(140, 21);
620.                 this.label13.TabIndex = 34;
621.                 this.label13.Text = "Numero de Carriles";
622.                 this.label13.Visible = false;
623.                 //
624.                 // ListLog
625.                 //
626.                 this.ListLog.Controls.Add(this.Asc);
627.                 this.ListLog.Controls.Add(this.Desc);
628.                 this.ListLog.Font = new System.Drawing.Font("Franklin Gothic
    Medium", 12F);
629.                 this.ListLog.ForeColor = System.Drawing.Color.White;
630.                 this.ListLog.Location = new System.Drawing.Point(782, 403);
631.                 this.ListLog.Margin = new System.Windows.Forms.Padding(2, 2, 2,
    2);
632.                 this.ListLog.Name = "ListLog";
633.                 this.ListLog.Padding = new System.Windows.Forms.Padding(2, 2, 2,
     2);
634.                 this.ListLog.Size = new System.Drawing.Size(193, 81);
635.                 this.ListLog.TabIndex = 39;
636.                 this.ListLog.TabStop = false;
637.                 this.ListLog.Text = "Seleccione Pendiente";
638.                 this.ListLog.Visible = false;
639.                 //
```

```
640.                 // Asc
641.                 //
642.                 this.Asc.AutoSize = true;
643.                 this.Asc.Location = new System.Drawing.Point(4, 46);
644.                 this.Asc.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
645.                 this.Asc.Name = "Asc";
646.                 this.Asc.Size = new System.Drawing.Size(108, 25);
647.                 this.Asc.TabIndex = 1;
648.                 this.Asc.TabStop = true;
649.                 this.Asc.Text = "Ascendente";
650.                 this.Asc.UseVisualStyleBackColor = true;
651.                 this.Asc.Visible = false;
652.                 this.Asc.CheckedChanged += new System.EventHandler(this.Asc_Chec
    kedChanged);
653.                 //
654.                 // Desc
655.                 //
656.                 this.Desc.AutoSize = true;
657.                 this.Desc.Location = new System.Drawing.Point(4, 16);
658.                 this.Desc.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
659.                 this.Desc.Name = "Desc";
660.                 this.Desc.Size = new System.Drawing.Size(116, 25);
661.                 this.Desc.TabIndex = 0;
662.                 this.Desc.TabStop = true;
663.                 this.Desc.Text = "Descendente";
664.                 this.Desc.UseVisualStyleBackColor = true;
665.                 this.Desc.Visible = false;
666.                 this.Desc.CheckedChanged += new System.EventHandler(this.Desc_Ch
    eckedChanged);
667.                 //
668.                 // Temporizador
669.                 //
670.                 this.Temporizador.Interval = 2000;
671.                 this.Temporizador.Tick += new System.EventHandler(this.Temporiza
    dor_Tick);
672.                 //
673.                 // openFileDialog1
674.                 //
675.                 this.openFileDialog1.FileName = "openFileDialog1";
676.                 //
677.                 // trackBar1
678.                 //
679.                 this.trackBar1.Location = new System.Drawing.Point(845, 118);
680.                 this.trackBar1.Margin = new System.Windows.Forms.Padding(2, 2, 2
    , 2);
681.                 this.trackBar1.Minimum = 1;
682.                 this.trackBar1.Name = "trackBar1";
683.                 this.trackBar1.Size = new System.Drawing.Size(130, 45);
684.                 this.trackBar1.TabIndex = 43;
685.                 this.trackBar1.Value = 1;
686.                 this.trackBar1.Visible = false;
687.                 this.trackBar1.ValueChanged += new System.EventHandler(this.trac
    kBar1_ValueChanged);
688.                 //
689.                 // bFinalizar
690.                 //
691.                 this.bFinalizar.Enabled = false;
692.                 this.bFinalizar.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
693.                 this.bFinalizar.ForeColor = System.Drawing.Color.FromArgb(((int)
    (((byte)(49)))), ((int)(((byte)(66)))), ((int)(((byte)(82)))));
```

```
694.                 this.bFinalizar.Image = global::ProjecteTFG.Properties.Resources
    .stop;
695.                 this.bFinalizar.Location = new System.Drawing.Point(757, 104);
696.                 this.bFinalizar.Margin = new System.Windows.Forms.Padding(2, 2,
    2, 2);
697.                 this.bFinalizar.Name = "bFinalizar";
698.                 this.bFinalizar.Size = new System.Drawing.Size(60, 59);
699.                 this.bFinalizar.TabIndex = 42;
700.                 this.bFinalizar.UseVisualStyleBackColor = true;
701.                 this.bFinalizar.Visible = false;
702.                 this.bFinalizar.Click += new System.EventHandler(this.bFinalizar
    _Click);
703.                 //
704.                 // bDetener
705.                 //
706.                 this.bDetener.Enabled = false;
707.                 this.bDetener.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
708.                 this.bDetener.ForeColor = System.Drawing.Color.FromArgb(((int)((
    (byte)(49)))), ((int)(((byte)(66)))), ((int)(((byte)(82)))));
709.                 this.bDetener.Image = global::ProjecteTFG.Properties.Resources.p
    ause;
710.                 this.bDetener.Location = new System.Drawing.Point(673, 104);
711.                 this.bDetener.Margin = new System.Windows.Forms.Padding(2, 2, 2,
     2);
712.                 this.bDetener.Name = "bDetener";
713.                 this.bDetener.Size = new System.Drawing.Size(60, 59);
714.                 this.bDetener.TabIndex = 41;
715.                 this.bDetener.UseVisualStyleBackColor = true;
716.                 this.bDetener.Visible = false;
717.                 this.bDetener.Click += new System.EventHandler(this.bDetener_Cli
    ck);
718.                 //
719.                 // bPlay
720.                 //
721.                 this.bPlay.Enabled = false;
722.                 this.bPlay.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
723.                 this.bPlay.ForeColor = System.Drawing.Color.FromArgb(((int)(((by
    te)(49)))), ((int)(((byte)(66)))), ((int)(((byte)(82)))));
724.                 this.bPlay.Image = global::ProjecteTFG.Properties.Resources.play
    ;
725.                 this.bPlay.Location = new System.Drawing.Point(586, 104);
726.                 this.bPlay.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2)
    ;
727.                 this.bPlay.Name = "bPlay";
728.                 this.bPlay.Size = new System.Drawing.Size(60, 59);
729.                 this.bPlay.TabIndex = 40;
730.                 this.bPlay.UseVisualStyleBackColor = true;
731.                 this.bPlay.Visible = false;
732.                 this.bPlay.Click += new System.EventHandler(this.bPlay_Click);
733.                 //
734.                 // btnCambio
735.                 //
736.                 this.btnCambio.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
737.                 this.btnCambio.Image = global::ProjecteTFG.Properties.Resources.
    Cambiar2;
738.                 this.btnCambio.Location = new System.Drawing.Point(241, 545);
739.                 this.btnCambio.Margin = new System.Windows.Forms.Padding(2, 2, 2
    , 2);
740.                 this.btnCambio.Name = "btnCambio";
741.                 this.btnCambio.Size = new System.Drawing.Size(35, 31);
742.                 this.btnCambio.TabIndex = 28;
```

```
743.                 this.btnCambio.UseVisualStyleBackColor = true;
744.                 this.btnCambio.Visible = false;
745.                 this.btnCambio.Click += new System.EventHandler(this.btnCambio_C
    lick);
746.                 //
747.                 // label5
748.                 //
749.                 this.label5.AutoSize = true;
750.                 this.label5.BackColor = System.Drawing.Color.White;
751.                 this.label5.Font = new System.Drawing.Font("Microsoft Sans
    Serif",
    10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
    (0)));
752.                 this.label5.Location = new System.Drawing.Point(625, 443);
753.                 this.label5.Margin = new System.Windows.Forms.Padding(2, 0, 2, 0
    );
754.                 this.label5.Name = "label5";
755.                 this.label5.Size = new System.Drawing.Size(76, 17);
756.                 this.label5.TabIndex = 44;
757.                 this.label5.Text = "Tiempo (s)";
758.                 this.label5.Visible = false;
759.                 //
760.                 // Form1
761.                 //
762.                 this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
763.                 this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
764.                 this.BackColor = System.Drawing.Color.FromArgb(((int)(((byte)(49
    )))), ((int)(((byte)(66)))), ((int)(((byte)(82)))));
765.                 this.ClientSize = new System.Drawing.Size(1071, 615);
766.                 this.Controls.Add(this.label5);
767.                 this.Controls.Add(this.trackBar1);
768.                 this.Controls.Add(this.bFinalizar);
769.                 this.Controls.Add(this.bDetener);
770.                 this.Controls.Add(this.bPlay);
771.                 this.Controls.Add(this.ListLog);
772.                 this.Controls.Add(this.label13);
773.                 this.Controls.Add(this.numerico3);
774.                 this.Controls.Add(this.Graf);
775.                 this.Controls.Add(this.comboDis);
776.                 this.Controls.Add(this.listaSalidas);
777.                 this.Controls.Add(this.label9);
778.                 this.Controls.Add(this.label10);
779.                 this.Controls.Add(this.listaIntermedios);
780.                 this.Controls.Add(this.label8);
781.                 this.Controls.Add(this.listaEntradas);
782.                 this.Controls.Add(this.btnCambio);
783.                 this.Controls.Add(this.LCuenta);
784.                 this.Controls.Add(this.LNombreProyecto);
785.                 this.Controls.Add(this.LInstruccion2);
786.                 this.Controls.Add(this.LInstruccion1);
787.                 this.Controls.Add(this.panel1);
788.                 this.Controls.Add(this.panel2);
789.                 this.Controls.Add(this.gMapControl1);
790.                 this.Controls.Add(this.dataGridView1);
791.                 this.Controls.Add(this.label12);
792.                 this.Controls.Add(this.numerico2);
793.                 this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixe
    dSingle;
794.                 this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Ic
    on")));
795.                 this.Margin = new System.Windows.Forms.Padding(2, 2, 2, 2);
```

```
796.                 this.MaximizeBox = false;
797.                 this.MinimizeBox = false;
798.                 this.Name = "Form1";
799.                 this.Text = "Simulador de Tráfico: TFG";
800.                 this.Load += new System.EventHandler(this.Form1_Load_1);
801.                 ((System.ComponentModel.ISupportInitialize)(this.dataGridView1))
    .EndInit();
802.                 this.panel2.ResumeLayout(false);
803.                 this.panel2.PerformLayout();
804.                 ((System.ComponentModel.ISupportInitialize)(this.numerico)).EndI
    nit();
805.                 this.panel1.ResumeLayout(false);
806.                 ((System.ComponentModel.ISupportInitialize)(this.Graf)).EndInit(
    );
807.                 ((System.ComponentModel.ISupportInitialize)(this.numerico2)).End
    Init();
808.                 ((System.ComponentModel.ISupportInitialize)(this.numerico3)).End
    Init();
809.                 this.ListLog.ResumeLayout(false);
810.                 this.ListLog.PerformLayout();
811.                 ((System.ComponentModel.ISupportInitialize)(this.trackBar1)).End
    Init();
812.                 this.ResumeLayout(false);
813.                 this.PerformLayout();
814.
815.             }
816.
817.             #endregion
818.
819.             private GMap.NET.WindowsForms.GMapControl gMapControl1;
820.             private System.Windows.Forms.Button btnAgregar;
821.             private System.Windows.Forms.TextBox txtdescripcion;
822.             private System.Windows.Forms.Label label1;
823.             private System.Windows.Forms.TextBox txtlatitud;
824.             private System.Windows.Forms.Label label2;
825.             private System.Windows.Forms.Label label3;
826.             private System.Windows.Forms.TextBox txtlongitud;
827.             private System.Windows.Forms.DataGridView dataGridView1;
828.             private System.Windows.Forms.Timer timer1;
829.             private System.Windows.Forms.Panel panel2;
830.             private System.Windows.Forms.TextBox textBox1;
831.             private System.Windows.Forms.Label LNewSim;
832.             private System.Windows.Forms.Panel panel1;
833.             private System.Windows.Forms.Button btnNewSim;
834.             private System.Windows.Forms.Label LInstruccion1;
835.             private System.Windows.Forms.Label LInstruccion2;
836.             private System.Windows.Forms.Button btnOk;
837.             private System.Windows.Forms.Label LNombreProyecto;
838.             private System.Windows.Forms.Label LCuenta;
839.             private System.Windows.Forms.Button btnCambio;
840.             private System.Windows.Forms.Label label6;
841.             private System.Windows.Forms.NumericUpDown numerico;
842.             private System.Windows.Forms.ListBox listaEntradas;
843.             private System.Windows.Forms.Label label8;
844.             private System.Windows.Forms.Label label9;
845.             private System.Windows.Forms.ListBox listaSalidas;
846.             private System.Windows.Forms.ListBox listaIntermedios;
847.             private System.Windows.Forms.Label label10;
848.             private System.Windows.Forms.ListBox tipoOut;
849.             private System.Windows.Forms.Label label11;
850.             private System.Windows.Forms.ComboBox comboDis;
```

```
851.             private System.Windows.Forms.DataVisualization.Charting.Chart Graf;
852.             private System.Windows.Forms.Label label12;
853.             private System.Windows.Forms.NumericUpDown numerico2;
854.             private System.Windows.Forms.NumericUpDown numerico3;
855.             private System.Windows.Forms.Label label13;
856.             private System.Windows.Forms.GroupBox ListLog;
857.             private System.Windows.Forms.RadioButton Asc;
858.             private System.Windows.Forms.RadioButton Desc;
859.             private System.Windows.Forms.Button bPlay;
860.             private System.Windows.Forms.Timer Temporizador;
861.             private System.Windows.Forms.Button bDetener;
862.             private System.Windows.Forms.Button bFinalizar;
863.             private System.Windows.Forms.FolderBrowserDialog folderBrowserDialog
    1;
864.             private System.Windows.Forms.Button btnLoadSim;
865.             private System.Windows.Forms.OpenFileDialog openFileDialog1;
866.             private System.Windows.Forms.SaveFileDialog saveFileDialog1;
867.             private System.Windows.Forms.TextBox txtVelMed;
868.             private System.Windows.Forms.Label label4;
869.             private System.Windows.Forms.TrackBar trackBar1;
870.             private System.Windows.Forms.Label label5;
871.         }
872.     }
```

Anexo IV: Archivo de Texto de los Resultados

"Resultados de la simulación del Proyecto: Ejemplo"

"\\\\\ TRAMO: Tramo1 \\\\"

Iteración,	Estado,	Vehículos,	Peatones,	Tiempo Restante,	Capacidades
0,	0,	0,	0,	14.4,	14
1,	0,	0,	0,	13.2,	14
2,	0,	0,	0,	12,	14
3,	0,	1,	0,	10.8,	14
4,	0,	2,	0,	9.6,	14
5,	0,	2,	0,	8.4,	14
6,	0,	2,	0,	7.2,	14
7,	0,	2,	0,	6,	14
8,	0,	2,	0,	4.8,	14
9,	0,	2,	0,	3.6,	14
10,	0,	2,	0,	2.4,	14
11,	0,	2,	0,	1.2,	14
12,	0,	2,	0,	0,	14
13,	0,	2,	0,	30,	14
14,	0,	3,	0,	28.8,	14
15,	0,	4,	0,	27.6,	14
16,	0,	4,	0,	26.4,	14
17,	0,	4,	0,	25.2,	14
18,	0,	4,	0,	24,	14
19,	0,	4,	0,	22.8,	14
20,	0,	4,	0,	21.6,	14
21,	0,	4,	0,	20.4,	14
22,	0,	4,	0,	19.2,	14
23,	0,	4,	0,	18,	14
24,	0,	4,	0,	16.8,	14
25,	0,	5,	0,	15.6,	14
26,	0,	6,	0,	14.4,	14
27,	0,	6,	0,	13.2,	14
28,	0,	6,	0,	12,	14
29,	0,	6,	0,	10.8,	14
30,	0,	6,	0,	9.6,	14
31,	0,	6,	0,	8.4,	14
32,	0,	6,	0,	7.2,	14
33,	0,	6,	0,	6,	14
34,	0,	6,	0,	4.8,	14
35,	0,	6,	0,	3.6,	14
36,	0,	7,	0,	2.4,	14
37,	0,	8,	0,	1.2,	14
38,	0,	8,	0,	0,	14
39,	1,	8,	0,	30,	14
40,	1,	6,	1,	28.8,	14
41,	1,	4,	1,	27.6,	14
42,	1,	3,	1,	26.4,	14
43,	1,	3,	1,	25.2,	14
44,	1,	2,	3,	24,	14
45,	1,	2,	5,	22.8,	14
46,	1,	2,	7,	21.6,	14
47,	1,	2,	8,	20.4,	14
48,	1,	2,	9,	19.2,	14
49,	1,	1,	10,	18,	14
50,	1,	0,	11,	16.8,	14
51,	1,	0,	12,	15.6,	14
52,	1,	0,	12,	14.4,	14
53,	1,	1,	12,	13.2,	14
54,	1,	2,	12,	12,	14
55,	1,	2,	14,	10.8,	14
56,	1,	2,	16,	9.6,	14

```
57,   1,   2,   18,   8.4,   14
58,   1,   2,   19,   7.2,   14
59,   1,   2,   20,   6,   14
60,   1,   1,   21,   4.8,   14
61,   1,   0,   22,   3.6,   14
62,   1,   0,   23,   2.4,   14
63,   1,   0,   23,   1.2,   14
64,   1,   1,   23,   0,   14
65,   0,   1,   23,   20.4,   14
66,   0,   1,   0,   19.2,   14
67,   0,   1,   0,   18,   14
68,   0,   1,   0,   16.8,   14
69,   0,   2,   0,   15.6,   14
70,   0,   3,   0,   14.4,   14
71,   0,   3,   0,   13.2,   14
72,   0,   3,   0,   12,   14
73,   0,   3,   0,   10.8,   14
74,   0,   3,   0,   9.6,   14
75,   0,   3,   0,   8.4,   14
76,   0,   3,   0,   7.2,   14
77,   0,   3,   0,   6,   14
78,   0,   3,   0,   4.8,   14
79,   0,   3,   0,   3.6,   14
80,   0,   4,   0,   2.4,   14
81,   0,   5,   0,   1.2,   14
82,   0,   5,   0,   0,   14
83,   0,   5,   0,   9.6,   14
84,   0,   5,   0,   8.4,   14
85,   0,   5,   0,   7.2,   14
86,   0,   5,   0,   6,   14
87,   0,   5,   0,   4.8,   14
88,   0,   5,   0,   3.6,   14
89,   0,   5,   0,   2.4,   14
90,   0,   5,   0,   1.2,   14
91,   0,   6,   0,   0,   14
92,   1,   7,   0,   30,   14
93,   1,   5,   1,   28.8,   14
94,   1,   3,   2,   27.6,   14
95,   1,   3,   3,   26.4,   14
96,   1,   4,   3,   25.2,   14
97,   1,   4,   3,   24,   14
98,   1,   3,   3,   22.8,   14
99,   1,   2,   5,   21.6,   14
100,   1,   2,   7,   20.4,   14
101,   1,   2,   9,   19.2,   14
102,   1,   1,   10,   18,   14
103,   1,   0,   11,   16.8,   14
104,   1,   0,   12,   15.6,   14
105,   1,   0,   13,   14.4,   14
106,   1,   1,   14,   13.2,   14
107,   1,   2,   14,   12,   14
108,   1,   2,   14,   10.8,   14
109,   1,   2,   14,   9.6,   14
110,   1,   2,   16,   8.4,   14
111,   1,   2,   18,   7.2,   14
112,   1,   2,   20,   6,   14
113,   1,   1,   21,   4.8,   14
114,   1,   0,   22,   3.6,   14
115,   1,   0,   23,   2.4,   14
116,   1,   0,   24,   1.2,   14
117,   1,   1,   25,   0,   14
118,   0,   1,   25,   30,   14
119,   0,   1,   0,   28.8,   14
120,   0,   1,   0,   27.6,   14
121,   0,   1,   0,   26.4,   14
122,   0,   1,   0,   25.2,   14
```

```
123,    0,    1,    0,    24,    14
124,    0,    2,    0,    22.8,    14
125,    0,    3,    0,    21.6,    14
126,    0,    3,    0,    20.4,    14
127,    0,    3,    0,    19.2,    14
128,    0,    3,    0,    18,    14
129,    0,    3,    0,    16.8,    14
130,    0,    3,    0,    15.6,    14
131,    0,    3,    0,    14.4,    14
132,    0,    3,    0,    13.2,    14
133,    0,    3,    0,    12,    14
134,    0,    3,    0,    10.8,    14
135,    0,    4,    0,    9.6,    14
136,    0,    5,    0,    8.4,    14
137,    0,    5,    0,    7.2,    14
138,    0,    5,    0,    6,    14
139,    0,    5,    0,    4.8,    14
140,    0,    5,    0,    3.6,    14
141,    0,    5,    0,    2.4,    14
142,    0,    5,    0,    1.2,    14
143,    0,    5,    0,    0,    14
144,    1,    5,    0,    30,    14
145,    1,    3,    2,    28.8,    14
146,    1,    1,    3,    27.6,    14
147,    1,    1,    4,    26.4,    14
148,    1,    2,    5,    25.2,    14
149,    1,    2,    6,    24,    14
150,    1,    2,    7,    22.8,    14
151,    1,    2,    7,    21.6,    14
152,    1,    2,    7,    20.4,    14
153,    1,    2,    7,    19.2,    14
154,    1,    1,    9,    18,    14
155,    1,    0,    11,    16.8,    14
156,    1,    0,    13,    15.6,    14
157,    1,    0,    14,    14.4,    14
158,    1,    1,    15,    13.2,    14
159,    1,    2,    16,    12,    14
160,    1,    2,    17,    10.8,    14
161,    1,    2,    18,    9.6,    14
162,    1,    2,    18,    8.4,    14
163,    1,    2,    18,    7.2,    14
164,    1,    2,    18,    6,    14
165,    1,    1,    20,    4.8,    14
166,    1,    0,    22,    3.6,    14
167,    1,    0,    24,    2.4,    14
168,    1,    0,    25,    1.2,    14
169,    1,    1,    26,    0,    14
170,    0,    1,    27,    30,    14
171,    0,    1,    0,    28.8,    14
172,    0,    1,    0,    27.6,    14
173,    0,    1,    0,    26.4,    14
174,    0,    1,    0,    25.2,    14
175,    0,    1,    0,    24,    14
176,    0,    1,    0,    22.8,    14
177,    0,    1,    0,    21.6,    14
178,    0,    1,    0,    20.4,    14
179,    0,    2,    0,    19.2,    14
180,    0,    3,    0,    18,    14
181,    0,    3,    0,    16.8,    14
182,    0,    3,    0,    15.6,    14
183,    0,    3,    0,    14.4,    14
184,    0,    3,    0,    13.2,    14
185,    0,    3,    0,    12,    14
186,    0,    3,    0,    10.8,    14
187,    0,    3,    0,    9.6,    14
188,    0,    3,    0,    8.4,    14
```

```
189,    0,    3,    0,    7.2,    14
190,    0,    4,    0,    6,    14
191,    0,    5,    0,    4.8,    14
192,    0,    5,    0,    3.6,    14
193,    0,    5,    0,    2.4,    14
194,    0,    5,    0,    1.2,    14
195,    0,    5,    0,    0,    14
196,    1,    5,    0,    30,    14
197,    1,    3,    0,    28.8,    14
198,    1,    1,    2,    27.6,    14
199,    1,    1,    4,    26.4,    14
200,    1,    2,    6,    25.2,    14
201,    1,    2,    7,    24,    14
202,    1,    2,    8,    22.8,    14
203,    1,    2,    9,    21.6,    14
204,    1,    2,    10,    20.4,    14
205,    1,    2,    11,    19.2,    14
206,    1,    1,    11,    18,    14
207,    1,    0,    11,    16.8,    14
208,    1,    0,    11,    15.6,    14
209,    1,    0,    13,    14.4,    14
210,    1,    1,    15,    13.2,    14
211,    1,    2,    17,    12,    14
212,    1,    2,    18,    10.8,    14
213,    1,    2,    19,    9.6,    14
214,    1,    2,    20,    8.4,    14
215,    1,    2,    21,    7.2,    14
216,    1,    2,    22,    6,    14
217,    1,    1,    22,    4.8,    14
218,    1,    0,    22,    3.6,    14
219,    1,    0,    22,    2.4,    14
220,    1,    0,    24,    1.2,    14
221,    1,    1,    26,    0,    14
222,    0,    1,    28,    30,    14
223,    0,    2,    0,    28.8,    14
224,    0,    3,    0,    27.6,    14
225,    0,    3,    0,    26.4,    14
226,    0,    3,    0,    25.2,    14
227,    0,    3,    0,    24,    14
228,    0,    3,    0,    22.8,    14
229,    0,    3,    0,    21.6,    14
230,    0,    3,    0,    20.4,    14
231,    0,    3,    0,    19.2,    14
232,    0,    3,    0,    18,    14
233,    0,    3,    0,    16.8,    14
234,    0,    4,    0,    15.6,    14
235,    0,    5,    0,    14.4,    14
236,    0,    5,    0,    13.2,    14
237,    0,    5,    0,    12,    14
238,    0,    5,    0,    10.8,    14
239,    0,    5,    0,    9.6,    14
240,    0,    5,    0,    8.4,    14
241,    0,    5,    0,    7.2,    14
242,    0,    5,    0,    6,    14
243,    0,    5,    0,    4.8,    14
244,    0,    5,    0,    3.6,    14
245,    0,    6,    0,    2.4,    14
246,    0,    7,    0,    1.2,    14
247,    0,    7,    0,    0,    14
248,    1,    7,    0,    30,    14
249,    1,    5,    1,    28.8,    14
250,    1,    3,    1,    27.6,    14
251,    1,    3,    1,    26.4,    14
252,    1,    3,    1,    25.2,    14
253,    1,    2,    3,    24,    14
254,    1,    2,    5,    22.8,    14
```

```
255,  1,  2,  7,  21.6,  14
256,  1,  2,  8,  20.4,  14
257,  1,  2,  9,  19.2,  14
258,  1,  1,  10,  18,  14
259,  1,  0,  11,  16.8,  14
260,  1,  0,  12,  15.6,  14
261,  1,  0,  12,  14.4,  14
262,  1,  1,  12,  13.2,  14
263,  1,  2,  12,  12,  14
264,  1,  2,  14,  10.8,  14
265,  1,  2,  16,  9.6,  14
266,  1,  2,  18,  8.4,  14
267,  1,  2,  19,  7.2,  14
268,  1,  2,  20,  6,  14
269,  1,  1,  21,  4.8,  14
270,  1,  0,  22,  3.6,  14
271,  1,  0,  23,  2.4,  14
272,  1,  0,  23,  1.2,  14
273,  1,  1,  23,  0,  14
274,  0,  1,  23,  30,  14
275,  0,  1,  0,  28.8,  14
276,  0,  1,  0,  27.6,  14
277,  0,  1,  0,  26.4,  14
278,  0,  2,  0,  25.2,  14
279,  0,  3,  0,  24,  14
280,  0,  3,  0,  22.8,  14
281,  0,  3,  0,  21.6,  14
282,  0,  3,  0,  20.4,  14
283,  0,  3,  0,  19.2,  14
284,  0,  3,  0,  18,  14
285,  0,  3,  0,  16.8,  14
286,  0,  3,  0,  15.6,  14
287,  0,  3,  0,  14.4,  14
288,  0,  3,  0,  13.2,  14
289,  0,  4,  0,  12,  14
290,  0,  5,  0,  10.8,  14
291,  0,  5,  0,  9.6,  14
292,  0,  5,  0,  8.4,  14
293,  0,  5,  0,  7.2,  14
294,  0,  5,  0,  6,  14
295,  0,  5,  0,  4.8,  14
296,  0,  5,  0,  3.6,  14
297,  0,  5,  0,  2.4,  14
298,  0,  5,  0,  1.2,  14
299,  0,  5,  0,  0,  14
300,  1,  6,  0,  30,  14
301,  1,  4,  1,  28.8,  14
302,  1,  2,  2,  27.6,  14
303,  1,  2,  3,  26.4,  14
304,  1,  3,  4,  25.2,  14
305,  1,  3,  4,  24,  14
306,  1,  3,  4,  22.8,  14
307,  1,  2,  4,  21.6,  14
308,  1,  2,  6,  20.4,  14
309,  1,  2,  8,  19.2,  14
310,  1,  1,  10,  18,  14
311,  1,  0,  11,  16.8,  14
312,  1,  0,  12,  15.6,  14
313,  1,  0,  13,  14.4,  14
314,  1,  1,  14,  13.2,  14
315,  1,  2,  15,  12,  14
316,  1,  2,  15,  10.8,  14
317,  1,  2,  15,  9.6,  14
318,  1,  2,  15,  8.4,  14
319,  1,  2,  17,  7.2,  14
320,  1,  2,  19,  6,  14
```

```
 321,    1,    1,    21,    4.8,    14
 322,    1,    0,    22,    3.6,    14
 323,    1,    0,    23,    2.4,    14
 324,    1,    0,    24,    1.2,    14
 325,    1,    1,    25,    0,    14
 326,    0,    1,    26,    30,    14
 327,    0,    1,    0,    28.8,    14
 328,    0,    1,    0,    27.6,    14
 329,    0,    1,    0,    26.4,    14
 330,    0,    1,    0,    25.2,    14
 331,    0,    1,    0,    24,    14
 332,    0,    1,    0,    22.8,    14
 333,    0,    2,    0,    21.6,    14
 334,    0,    3,    0,    20.4,    14
 335,    0,    3,    0,    19.2,    14
 336,    0,    3,    0,    18,    14
 337,    0,    3,    0,    16.8,    14
 338,    0,    3,    0,    15.6,    14
 339,    0,    3,    0,    14.4,    14
 340,    0,    3,    0,    13.2,    14
 341,    0,    3,    0,    12,    14
 342,    0,    3,    0,    10.8,    14
 343,    0,    3,    0,    9.6,    14
 344,    0,    4,    0,    8.4,    14
 345,    0,    5,    0,    7.2,    14
 346,    0,    5,    0,    6,    14
 347,    0,    5,    0,    4.8,    14
 348,    0,    5,    0,    3.6,    14
 349,    0,    5,    0,    2.4,    14
 350,    0,    5,    0,    1.2,    14
 351,    0,    5,    0,    0,    14
 352,    1,    5,    0,    30,    14
 353,    1,    3,    2,    28.8,    14
 354,    1,    1,    4,    27.6,    14
 355,    1,    1,    5,    26.4,    14
 356,    1,    2,    6,    25.2,    14
 357,    1,    2,    7,    24,    14
 358,    1,    2,    8,    22.8,    14
 359,    1,    2,    9,    21.6,    14
 360,    1,    2,    9,    20.4,    14
 361,    1,    2,    9,    19.2,    14
 362,    1,    1,    9,    18,    14
 363,    1,    0,    11,    16.8,    14
 364,    1,    0,    13,    15.6,    14
 365,    1,    0,    15,    14.4,    14
 366,    1,    1,    16,    13.2,    14
 367,    1,    2,    17,    12,    14
 368,    1,    2,    18,    10.8,    14
 369,    1,    2,    19,    9.6,    14
 370,    1,    2,    20,    8.4,    14
 371,    1,    2,    20,    7.2,    14
 372,    1,    2,    20,    6,    14
 373,    1,    1,    20,    4.8,    14
 374,    1,    0,    22,    3.6,    14
 375,    1,    0,    24,    2.4,    14
 376,    1,    0,    26,    1.2,    14
 377,    1,    1,    27,    0,    14
 378,    0,    2,    28,    30,    14
 379,    0,    2,    0,    28.8,    14
 380,    0,    2,    0,    27.6,    14
 381,    0,    2,    0,    26.4,    14
 382,    0,    2,    0,    25.2,    14
 383,    0,    2,    0,    24,    14
 384,    0,    2,    0,    22.8,    14
 385,    0,    2,    0,    21.6,    14
 386,    0,    2,    0,    20.4,    14
```

```
    387,        0,        2,        0,        19.2,        14
    388,        0,        3,        0,        18,        14
    389,        0,        4,        0,        16.8,        14
    390,        0,        4,        0,        15.6,        14
    391,        0,        4,        0,        14.4,        14
    392,        0,        4,        0,        13.2,        14
    393,        0,        4,        0,        12,        14
    394,        0,        4,        0,        10.8,        14
    395,        0,        4,        0,        9.6,        14
    396,        0,        4,        0,        8.4,        14
    397,        0,        4,        0,        7.2,        14
    398,        0,        4,        0,        6,        14
    399,        0,        5,        0,        4.8,        14
    400,        0,        6,        0,        3.6,        14
    401,        0,        6,        0,        2.4,        14
    402,        0,        6,        0,        1.2,        14
    403,        0,        6,        0,        0,        14
    404,        1,        6,        0,        30,        14
    405,        1,        4,        0,        28.8,        14
    406,        1,        2,        0,        27.6,        14
    407,        1,        1,        2,        26.4,        14
    408,        1,        2,        4,        25.2,        14
    409,        1,        2,        6,        24,        14
    410,        1,        2,        7,        22.8,        14
    411,        1,        2,        8,        21.6,        14
    412,        1,        2,        9,        20.4,        14
    413,        1,        2,        10,        19.2,        14
    414,        1,        1,        11,        18,        14
    415,        1,        0,        11,        16.8,        14
    416,        1,        0,        11,        15.6,        14
    417,        1,        0,        11,        14.4,        14
    418,        1,        1,        13,        13.2,        14
    419,        1,        2,        15,        12,        14
    420,        1,        2,        17,        10.8,        14
    421,        1,        2,        18,        9.6,        14
    422,        1,        2,        19,        8.4,        14
    423,        1,        2,        20,        7.2,        14
    424,        1,        2,        21,        6,        14
    425,        1,        1,        22,        4.8,        14
    426,        1,        0,        22,        3.6,        14
    427,        1,        0,        22,        2.4,        14
    428,        1,        0,        22,        1.2,        14
    429,        1,        1,        24,        0,        14
    430,        0,        1,        26,        30,        14
    431,        0,        1,        0,        28.8,        14
    432,        0,        2,        0,        27.6,        14
    433,        0,        3,        0,        26.4,        14
    434,        0,        3,        0,        25.2,        14
    435,        0,        3,        0,        24,        14
    436,        0,        3,        0,        22.8,        14
    437,        0,        3,        0,        21.6,        14
    438,        0,        3,        0,        20.4,        14
    439,        0,        3,        0,        19.2,        14
    440,        0,        3,        0,        18,        14
    441,        0,        3,        0,        16.8,        14
    442,        0,        3,        0,        15.6,        14
    443,        0,        4,        0,        14.4,        14
    444,        0,        5,        0,        13.2,        14
    445,        0,        5,        0,        12,        14
    446,        0,        5,        0,        10.8,        14
    447,        0,        5,        0,        9.6,        14
    448,        0,        5,        0,        8.4,        14
    449,        0,        5,        0,        7.2,        14
    450,        0,        5,        0,        6,        14
    451,        0,        5,        0,        4.8,        14
    452,        0,        5,        0,        3.6,        14
```

```
453,    0,    5,    0,    2.4,    14
454,    0,    6,    0,    1.2,    14
455,    0,    7,    0,    0,    14
456,    1,    7,    0,    30,    14
457,    1,    5,    1,    28.8,    14
458,    1,    3,    2,    27.6,    14
459,    1,    3,    2,    26.4,    14
460,    1,    4,    2,    25.2,    14
461,    1,    3,    2,    24,    14
462,    1,    2,    4,    22.8,    14
463,    1,    2,    6,    21.6,    14
464,    1,    2,    8,    20.4,    14
465,    1,    2,    9,    19.2,    14
466,    1,    1,    10,    18,    14
467,    1,    0,    11,    16.8,    14
468,    1,    0,    12,    15.6,    14
469,    1,    0,    13,    14.4,    14
470,    1,    1,    13,    13.2,    14
471,    1,    2,    13,    12,    14
472,    1,    2,    13,    10.8,    14
473,    1,    2,    15,    9.6,    14
474,    1,    2,    17,    8.4,    14
475,    1,    2,    19,    7.2,    14
476,    1,    2,    20,    6,    14
477,    1,    1,    21,    4.8,    14
478,    1,    0,    22,    3.6,    14
479,    1,    0,    23,    2.4,    14
480,    1,    0,    24,    1.2,    14
481,    1,    1,    24,    0,    14
482,    0,    1,    24,    30,    14
483,    0,    1,    0,    28.8,    14
484,    0,    1,    0,    27.6,    14
485,    0,    1,    0,    26.4,    14
486,    0,    1,    0,    25.2,    14
487,    0,    2,    0,    24,    14
488,    0,    3,    0,    22.8,    14
489,    0,    3,    0,    21.6,    14
490,    0,    3,    0,    20.4,    14
491,    0,    3,    0,    19.2,    14
492,    0,    3,    0,    18,    14
493,    0,    3,    0,    16.8,    14
494,    0,    3,    0,    15.6,    14
495,    0,    3,    0,    14.4,    14
496,    0,    3,    0,    13.2,    14
497,    0,    3,    0,    12,    14
498,    0,    4,    0,    10.8,    14
499,    0,    5,    0,    9.6,    14
500,    0,    5,    0,    8.4,    14
501,    0,    5,    0,    7.2,    14
502,    0,    5,    0,    6,    14
503,    0,    5,    0,    4.8,    14
504,    0,    5,    0,    3.6,    14
505,    0,    5,    0,    2.4,    14
506,    0,    5,    0,    1.2,    14
507,    0,    5,    0,    0,    14
508,    1,    5,    0,    30,    14
509,    1,    3,    1,    28.8,    14
510,    1,    1,    2,    27.6,    14
511,    1,    1,    3,    26.4,    14
512,    1,    2,    4,    25.2,    14
513,    1,    2,    5,    24,    14
514,    1,    2,    5,    22.8,    14
515,    1,    2,    5,    21.6,    14
516,    1,    2,    5,    20.4,    14
517,    1,    2,    7,    19.2,    14
518,    1,    1,    9,    18,    14
```

```
519,    1,    0,    11,    16.8,    14
520,    1,    0,    12,    15.6,    14
521,    1,    0,    13,    14.4,    14
522,    1,    1,    14,    13.2,    14
523,    1,    2,    15,    12,    14
524,    1,    2,    16,    10.8,    14
525,    1,    2,    16,    9.6,    14
526,    1,    2,    16,    8.4,    14
527,    1,    2,    16,    7.2,    14
528,    1,    2,    18,    6,    14
529,    1,    1,    20,    4.8,    14
530,    1,    0,    22,    3.6,    14
531,    1,    0,    23,    2.4,    14
532,    1,    0,    24,    1.2,    14
533,    1,    1,    25,    0,    14
534,    0,    1,    26,    30,    14
535,    0,    1,    0,    28.8,    14
536,    0,    1,    0,    27.6,    14
537,    0,    1,    0,    26.4,    14
538,    0,    1,    0,    25.2,    14
539,    0,    1,    0,    24,    14
540,    0,    1,    0,    22.8,    14
541,    0,    1,    0,    21.6,    14
542,    0,    2,    0,    20.4,    14
543,    0,    3,    0,    19.2,    14
544,    0,    3,    0,    18,    14
545,    0,    3,    0,    16.8,    14
546,    0,    3,    0,    15.6,    14
547,    0,    3,    0,    14.4,    14
548,    0,    3,    0,    13.2,    14
549,    0,    3,    0,    12,    14
550,    0,    3,    0,    10.8,    14
551,    0,    3,    0,    9.6,    14
552,    0,    3,    0,    8.4,    14
553,    0,    4,    0,    7.2,    14
554,    0,    5,    0,    6,    14
555,    0,    5,    0,    4.8,    14
556,    0,    5,    0,    3.6,    14
557,    0,    5,    0,    2.4,    14
558,    0,    5,    0,    1.2,    14
559,    0,    5,    0,    0,    14
560,    1,    5,    0,    30,    14
561,    1,    3,    2,    28.8,    14
562,    1,    1,    4,    27.6,    14
563,    1,    1,    6,    26.4,    14
564,    1,    2,    7,    25.2,    14
565,    1,    2,    8,    24,    14
566,    1,    2,    9,    22.8,    14
567,    1,    2,    10,    21.6,    14
568,    1,    2,    11,    20.4,    14
569,    1,    2,    11,    19.2,    14
570,    1,    1,    11,    18,    14
571,    1,    0,    11,    16.8,    14
572,    1,    0,    13,    15.6,    14
573,    1,    0,    15,    14.4,    14
574,    1,    1,    17,    13.2,    14
575,    1,    2,    18,    12,    14
576,    1,    2,    19,    10.8,    14
577,    1,    2,    20,    9.6,    14
578,    1,    2,    21,    8.4,    14
579,    1,    2,    22,    7.2,    14
580,    1,    2,    22,    6,    14
581,    1,    1,    22,    4.8,    14
582,    1,    0,    22,    3.6,    14
583,    1,    0,    24,    2.4,    14
584,    1,    0,    26,    1.2,    14
```

```
585,    1,    1,    28,    0,    14
586,    0,    2,    29,    30,    14
587,    0,    3,    0,    28.8,    14
588,    0,    3,    0,    27.6,    14
589,    0,    3,    0,    26.4,    14
590,    0,    3,    0,    25.2,    14
591,    0,    3,    0,    24,    14
592,    0,    3,    0,    22.8,    14
593,    0,    3,    0,    21.6,    14
594,    0,    3,    0,    20.4,    14
595,    0,    3,    0,    19.2,    14
596,    0,    3,    0,    18,    14
597,    0,    4,    0,    16.8,    14
598,    0,    5,    0,    15.6,    14
599,    0,    5,    0,    14.4,    14
600,    0,    5,    0,    13.2,    14
601,    0,    5,    0,    12,    14
602,    0,    5,    0,    10.8,    14
603,    0,    5,    0,    9.6,    14
604,    0,    5,    0,    8.4,    14
605,    0,    5,    0,    7.2,    14
606,    0,    5,    0,    6,    14
607,    0,    5,    0,    4.8,    14
608,    0,    6,    0,    3.6,    14
609,    0,    7,    0,    2.4,    14
610,    0,    7,    0,    1.2,    14
611,    0,    7,    0,    0,    14
612,    1,    7,    0,    30,    14
613,    1,    5,    0,    28.8,    14
614,    1,    3,    0,    27.6,    14
615,    1,    2,    0,    26.4,    14
616,    1,    2,    2,    25.2,    14
617,    1,    2,    4,    24,    14
618,    1,    2,    6,    22.8,    14
619,    1,    2,    7,    21.6,    14
620,    1,    2,    8,    20.4,    14
621,    1,    2,    9,    19.2,    14
622,    1,    1,    10,    18,    14
623,    1,    0,    11,    16.8,    14
624,    1,    0,    11,    15.6,    14
625,    1,    0,    11,    14.4,    14
626,    1,    1,    11,    13.2,    14
627,    1,    2,    13,    12,    14
628,    1,    2,    15,    10.8,    14
629,    1,    2,    17,    9.6,    14
630,    1,    2,    18,    8.4,    14
631,    1,    2,    19,    7.2,    14
632,    1,    2,    20,    6,    14
633,    1,    1,    21,    4.8,    14
634,    1,    0,    22,    3.6,    14
635,    1,    0,    22,    2.4,    14
636,    1,    0,    22,    1.2,    14
637,    1,    1,    22,    0,    14
638,    0,    1,    24,    30,    14
639,    0,    1,    0,    28.8,    14
640,    0,    1,    0,    27.6,    14
641,    0,    2,    0,    26.4,    14
642,    0,    3,    0,    25.2,    14
643,    0,    3,    0,    24,    14
644,    0,    3,    0,    22.8,    14
645,    0,    3,    0,    21.6,    14
646,    0,    3,    0,    20.4,    14
647,    0,    3,    0,    19.2,    14
648,    0,    3,    0,    18,    14
649,    0,    3,    0,    16.8,    14
650,    0,    3,    0,    15.6,    14
```

```
651,    0,    3,    0,    14.4,    14
652,    0,    4,    0,    13.2,    14
653,    0,    5,    0,    12,    14
654,    0,    5,    0,    10.8,    14
655,    0,    5,    0,    9.6,    14
656,    0,    5,    0,    8.4,    14
657,    0,    5,    0,    7.2,    14
658,    0,    5,    0,    6,    14
659,    0,    5,    0,    4.8,    14
660,    0,    5,    0,    3.6,    14
661,    0,    5,    0,    2.4,    14
662,    0,    5,    0,    1.2,    14
663,    0,    6,    0,    0,    14
664,    1,    7,    0,    30,    14
665,    1,    5,    1,    28.8,    14
666,    1,    3,    2,    27.6,    14
667,    1,    3,    3,    26.4,    14
668,    1,    4,    3,    25.2,    14
669,    1,    4,    3,    24,    14
670,    1,    3,    3,    22.8,    14
671,    1,    2,    5,    21.6,    14
672,    1,    2,    7,    20.4,    14
673,    1,    2,    9,    19.2,    14
674,    1,    1,    10,    18,    14
675,    1,    0,    11,    16.8,    14
676,    1,    0,    12,    15.6,    14
677,    1,    0,    13,    14.4,    14
678,    1,    1,    14,    13.2,    14
679,    1,    2,    14,    12,    14
680,    1,    2,    14,    10.8,    14
681,    1,    2,    14,    9.6,    14
682,    1,    2,    16,    8.4,    14
683,    1,    2,    18,    7.2,    14
684,    1,    2,    20,    6,    14
685,    1,    1,    21,    4.8,    14
686,    1,    0,    22,    3.6,    14
687,    1,    0,    23,    2.4,    14
688,    1,    0,    24,    1.2,    14
689,    1,    1,    25,    0,    14
690,    0,    1,    25,    30,    14
691,    0,    1,    0,    28.8,    14
692,    0,    1,    0,    27.6,    14
693,    0,    1,    0,    26.4,    14
694,    0,    1,    0,    25.2,    14
695,    0,    1,    0,    24,    14
696,    0,    2,    0,    22.8,    14
697,    0,    3,    0,    21.6,    14
698,    0,    3,    0,    20.4,    14
699,    0,    3,    0,    19.2,    14
700,    0,    3,    0,    18,    14
701,    0,    3,    0,    16.8,    14
702,    0,    3,    0,    15.6,    14
703,    0,    3,    0,    14.4,    14
704,    0,    3,    0,    13.2,    14
705,    0,    3,    0,    12,    14
706,    0,    3,    0,    10.8,    14
707,    0,    4,    0,    9.6,    14
708,    0,    5,    0,    8.4,    14
709,    0,    5,    0,    7.2,    14
710,    0,    5,    0,    6,    14
711,    0,    5,    0,    4.8,    14
712,    0,    5,    0,    3.6,    14
713,    0,    5,    0,    2.4,    14
714,    0,    5,    0,    1.2,    14
715,    0,    5,    0,    0,    14
716,    1,    5,    0,    30,    14
```

```
     717,              1,              3,              2,              28.8,              14
     718,              1,              1,              3,              27.6,              14
     719,              1,              1,              4,              26.4,              14
     720,              1,              2,              5,              25.2,              14
     721,              1,              2,              6,              24,              14
     722,              1,              2,              7,              22.8,              14
     723,              1,              2,              7,              21.6,              14
     724,              1,              2,              7,              20.4,              14
     725,              1,              2,              7,              19.2,              14
     726,              1,              1,              9,              18,              14
     727,              1,              0,              11,              16.8,              14
     728,              1,              0,              13,              15.6,              14
     729,              1,              0,              14,              14.4,              14
     730,              1,              1,              15,              13.2,              14
     731,              1,              2,              16,              12,              14
     732,              1,              2,              17,              10.8,              14
     733,              1,              2,              18,              9.6,              14
     734,              1,              2,              18,              8.4,              14
     735,              1,              2,              18,              7.2,              14
     736,              1,              2,              18,              6,              14
     737,              1,              1,              20,              4.8,              14
     738,              1,              0,              22,              3.6,              14
     739,              1,              0,              24,              2.4,              14
     740,              1,              0,              25,              1.2,              14
     741,              1,              1,              26,              0,              14
     742,              0,              1,              27,              30,              14
     743,              0,              1,              0,              28.8,              14
     744,              0,              1,              0,              27.6,              14
     745,              0,              1,              0,              26.4,              14
     746,              0,              1,              0,              25.2,              14

"\\\\\ TRAMO: Tramo2 \\\\"
Iteracion,          Estado,          Vehículos,          Peatones,          Tiempo Restante,          Capacidades
     0,              0,              0,              0,              13.909,              55
     1,              0,              0,              0,              13.091,              55
     2,              0,              1,              0,              12.273,              55
     3,              0,              3,              0,              11.455,              55
     4,              0,              4,              0,              10.636,              55
     5,              0,              4,              0,              9.818,              55
     6,              0,              4,              0,              9,              55
     7,              0,              4,              0,              8.182,              55
     8,              0,              4,              0,              7.364,              55
     9,              0,              4,              0,              6.545,              55
     10,              0,              4,              0,              5.727,              55
     11,              0,              4,              0,              4.909,              55
     12,              0,              4,              0,              4.091,              55
     13,              0,              5,              0,              3.273,              55
     14,              0,              7,              0,              2.455,              55
     15,              0,              8,              0,              1.636,              55
     16,              0,              8,              0,              0.818,              55
     17,              0,              8,              0,              0,              55
     18,              1,              8,              0,              30.273,              55
     19,              1,              8,              0,              29.455,              55
     20,              1,              7,              0,              28.636,              55
     21,              1,              6,              0,              27.818,              55
     22,              1,              7,              0,              27,              55
     23,              1,              8,              0,              26.182,              55
     24,              1,              8,              1,              25.364,              55
     25,              1,              8,              3,              24.545,              55
     26,              1,              8,              4,              23.727,              55
     27,              1,              8,              4,              22.909,              55
     28,              1,              8,              4,              22.091,              55
     29,              1,              8,              4,              21.273,              55
     30,              1,              8,              4,              20.455,              55
     31,              1,              7,              4,              19.636,              55
     32,              1,              6,              4,              18.818,              55
```

```
33,  1,  7,  4,  18,  55
34,  1,  8,  4,  17.182,  55
35,  1,  8,  5,  16.364,  55
36,  1,  8,  7,  15.545,  55
37,  1,  8,  8,  14.727,  55
38,  1,  8,  8,  13.909,  55
39,  1,  7,  8,  13.091,  55
40,  1,  5,  8,  12.273,  55
41,  1,  4,  8,  11.455,  55
42,  1,  4,  8,  10.636,  55
43,  1,  5,  8,  9.818,  55
44,  1,  7,  8,  9,  55
45,  1,  8,  8,  8.182,  55
46,  1,  8,  9,  7.364,  55
47,  1,  8,  11,  6.545,  55
48,  1,  8,  12,  5.727,  55
49,  1,  8,  12,  4.909,  55
50,  1,  7,  12,  4.091,  55
51,  1,  5,  12,  3.273,  55
52,  1,  4,  12,  2.455,  55
53,  1,  4,  12,  1.636,  55
54,  1,  5,  12,  0.818,  55
55,  1,  7,  12,  0,  55
56,  0,  7,  12,  19.636,  55
57,  0,  8,  0,  18.818,  55
58,  0,  10,  0,  18,  55
59,  0,  11,  0,  17.182,  55
60,  0,  11,  0,  16.364,  55
61,  0,  11,  0,  15.545,  55
62,  0,  11,  0,  14.727,  55
63,  0,  11,  0,  13.909,  55
64,  0,  11,  0,  13.091,  55
65,  0,  11,  0,  12.273,  55
66,  0,  11,  0,  11.455,  55
67,  0,  11,  0,  10.636,  55
68,  0,  12,  0,  9.818,  55
69,  0,  14,  0,  9,  55
70,  0,  15,  0,  8.182,  55
71,  0,  15,  0,  7.364,  55
72,  0,  15,  0,  6.545,  55
73,  0,  15,  0,  5.727,  55
74,  0,  15,  0,  4.909,  55
75,  0,  15,  0,  4.091,  55
76,  0,  15,  0,  3.273,  55
77,  0,  15,  0,  2.455,  55
78,  0,  15,  0,  1.636,  55
79,  0,  16,  0,  0.818,  55
80,  0,  18,  0,  0,  55
81,  0,  19,  0,  10.636,  55
82,  0,  19,  0,  9.818,  55
83,  0,  19,  0,  9,  55
84,  0,  19,  0,  8.182,  55
85,  0,  19,  0,  7.364,  55
86,  0,  19,  0,  6.545,  55
87,  0,  19,  0,  5.727,  55
88,  0,  19,  0,  4.909,  55
89,  0,  19,  0,  4.091,  55
90,  0,  20,  0,  3.273,  55
91,  0,  22,  0,  2.455,  55
92,  0,  23,  0,  1.636,  55
93,  0,  23,  0,  0.818,  55
94,  0,  23,  0,  0,  55
95,  1,  23,  0,  30.273,  55
96,  1,  20,  0,  29.455,  55
97,  1,  17,  0,  28.636,  55
98,  1,  15,  0,  27.818,  55
```

```
99,   1,   14,   0,   27,   55
100,   1,   12,   0,   26.182,   55
101,   1,   9,   1,   25.364,   55
102,   1,   8,   3,   24.545,   55
103,   1,   8,   4,   23.727,   55
104,   1,   8,   4,   22.909,   55
105,   1,   8,   4,   22.091,   55
106,   1,   8,   4,   21.273,   55
107,   1,   8,   4,   20.455,   55
108,   1,   7,   4,   19.636,   55
109,   1,   6,   4,   18.818,   55
110,   1,   7,   4,   18,   55
111,   1,   8,   4,   17.182,   55
112,   1,   8,   5,   16.364,   55
113,   1,   8,   7,   15.545,   55
114,   1,   8,   8,   14.727,   55
115,   1,   8,   8,   13.909,   55
116,   1,   7,   8,   13.091,   55
117,   1,   5,   8,   12.273,   55
118,   1,   4,   8,   11.455,   55
119,   1,   4,   8,   10.636,   55
120,   1,   5,   8,   9.818,   55
121,   1,   7,   8,   9,   55
122,   1,   8,   8,   8.182,   55
123,   1,   8,   9,   7.364,   55
124,   1,   8,   11,   6.545,   55
125,   1,   8,   12,   5.727,   55
126,   1,   8,   12,   4.909,   55
127,   1,   7,   12,   4.091,   55
128,   1,   5,   12,   3.273,   55
129,   1,   4,   12,   2.455,   55
130,   1,   4,   12,   1.636,   55
131,   1,   5,   12,   0.818,   55
132,   1,   7,   12,   0,   55
133,   0,   7,   12,   30.273,   55
134,   0,   8,   0,   29.455,   55
135,   0,   10,   0,   28.636,   55
136,   0,   11,   0,   27.818,   55
137,   0,   11,   0,   27,   55
138,   0,   11,   0,   26.182,   55
139,   0,   11,   0,   25.364,   55
140,   0,   11,   0,   24.545,   55
141,   0,   11,   0,   23.727,   55
142,   0,   11,   0,   22.909,   55
143,   0,   11,   0,   22.091,   55
144,   0,   11,   0,   21.273,   55
145,   0,   12,   0,   20.455,   55
146,   0,   14,   0,   19.636,   55
147,   0,   15,   0,   18.818,   55
148,   0,   15,   0,   18,   55
149,   0,   15,   0,   17.182,   55
150,   0,   15,   0,   16.364,   55
151,   0,   15,   0,   15.545,   55
152,   0,   15,   0,   14.727,   55
153,   0,   15,   0,   13.909,   55
154,   0,   15,   0,   13.091,   55
155,   0,   15,   0,   12.273,   55
156,   0,   16,   0,   11.455,   55
157,   0,   18,   0,   10.636,   55
158,   0,   19,   0,   9.818,   55
159,   0,   19,   0,   9,   55
160,   0,   19,   0,   8.182,   55
161,   0,   19,   0,   7.364,   55
162,   0,   19,   0,   6.545,   55
163,   0,   19,   0,   5.727,   55
164,   0,   19,   0,   4.909,   55
```

```
    165,          0,          19,          0,          4.091,          55
    166,          0,          19,          0,          3.273,          55
    167,          0,          20,          0,          2.455,          55
    168,          0,          22,          0,          1.636,          55
    169,          0,          23,          0,          0.818,          55
    170,          0,          23,          0,          0,          55
    171,          1,          23,          0,          30.273,          55
    172,          1,          20,          0,          29.455,          55
    173,          1,          17,          0,          28.636,          55
    174,          1,          15,          0,          27.818,          55
    175,          1,          14,          0,          27,          55
    176,          1,          12,          0,          26.182,          55
    177,          1,          9,          0,          25.364,          55
    178,          1,          8,          1,          24.545,          55
    179,          1,          8,          3,          23.727,          55
    180,          1,          8,          4,          22.909,          55
    181,          1,          8,          4,          22.091,          55
    182,          1,          8,          4,          21.273,          55
    183,          1,          8,          4,          20.455,          55
    184,          1,          8,          4,          19.636,          55
    185,          1,          8,          4,          18.818,          55
    186,          1,          8,          4,          18,          55
    187,          1,          8,          4,          17.182,          55
    188,          1,          8,          4,          16.364,          55
    189,          1,          8,          5,          15.545,          55
    190,          1,          8,          7,          14.727,          55
    191,          1,          8,          8,          13.909,          55
    192,          1,          7,          8,          13.091,          55
    193,          1,          5,          8,          12.273,          55
    194,          1,          4,          8,          11.455,          55
    195,          1,          4,          8,          10.636,          55
    196,          1,          5,          8,          9.818,          55
    197,          1,          7,          8,          9,          55
    198,          1,          8,          8,          8.182,          55
    199,          1,          8,          8,          7.364,          55
    200,          1,          8,          9,          6.545,          55
    201,          1,          8,          11,          5.727,          55
    202,          1,          8,          12,          4.909,          55
    203,          1,          7,          12,          4.091,          55
    204,          1,          5,          12,          3.273,          55
    205,          1,          4,          12,          2.455,          55
    206,          1,          4,          12,          1.636,          55
    207,          1,          5,          12,          0.818,          55
    208,          1,          7,          12,          0,          55
    209,          0,          7,          12,          30.273,          55
    210,          0,          7,          0,          29.455,          55
    211,          0,          8,          0,          28.636,          55
    212,          0,          10,          0,          27.818,          55
    213,          0,          11,          0,          27,          55
    214,          0,          11,          0,          26.182,          55
    215,          0,          11,          0,          25.364,          55
    216,          0,          11,          0,          24.545,          55
    217,          0,          11,          0,          23.727,          55
    218,          0,          11,          0,          22.909,          55
    219,          0,          11,          0,          22.091,          55
    220,          0,          11,          0,          21.273,          55
    221,          0,          11,          0,          20.455,          55
    222,          0,          12,          0,          19.636,          55
    223,          0,          14,          0,          18.818,          55
    224,          0,          15,          0,          18,          55
    225,          0,          15,          0,          17.182,          55
    226,          0,          15,          0,          16.364,          55
    227,          0,          15,          0,          15.545,          55
    228,          0,          15,          0,          14.727,          55
    229,          0,          15,          0,          13.909,          55
    230,          0,          15,          0,          13.091,          55
```

```
231,    0,    15,    0,    12.273,    55
232,    0,    15,    0,    11.455,    55
233,    0,    16,    0,    10.636,    55
234,    0,    18,    0,    9.818,    55
235,    0,    19,    0,    9,    55
236,    0,    19,    0,    8.182,    55
237,    0,    19,    0,    7.364,    55
238,    0,    19,    0,    6.545,    55
239,    0,    19,    0,    5.727,    55
240,    0,    19,    0,    4.909,    55
241,    0,    19,    0,    4.091,    55
242,    0,    19,    0,    3.273,    55
243,    0,    19,    0,    2.455,    55
244,    0,    20,    0,    1.636,    55
245,    0,    22,    0,    0.818,    55
246,    0,    23,    0,    0,    55
247,    1,    23,    0,    30.273,    55
248,    1,    20,    0,    29.455,    55
249,    1,    17,    0,    28.636,    55
250,    1,    15,    0,    27.818,    55
251,    1,    14,    0,    27,    55
252,    1,    12,    0,    26.182,    55
253,    1,    9,    0,    25.364,    55
254,    1,    8,    0,    24.545,    55
255,    1,    8,    1,    23.727,    55
256,    1,    8,    3,    22.909,    55
257,    1,    8,    4,    22.091,    55
258,    1,    8,    4,    21.273,    55
259,    1,    8,    4,    20.455,    55
260,    1,    8,    4,    19.636,    55
261,    1,    9,    4,    18.818,    55
262,    1,    10,    4,    18,    55
263,    1,    9,    4,    17.182,    55
264,    1,    8,    4,    16.364,    55
265,    1,    8,    4,    15.545,    55
266,    1,    8,    5,    14.727,    55
267,    1,    8,    7,    13.909,    55
268,    1,    7,    8,    13.091,    55
269,    1,    5,    8,    12.273,    55
270,    1,    4,    8,    11.455,    55
271,    1,    4,    8,    10.636,    55
272,    1,    5,    8,    9.818,    55
273,    1,    7,    8,    9,    55
274,    1,    8,    8,    8.182,    55
275,    1,    8,    8,    7.364,    55
276,    1,    8,    8,    6.545,    55
277,    1,    8,    9,    5.727,    55
278,    1,    8,    11,    4.909,    55
279,    1,    7,    12,    4.091,    55
280,    1,    5,    12,    3.273,    55
281,    1,    4,    12,    2.455,    55
282,    1,    4,    12,    1.636,    55
283,    1,    5,    12,    0.818,    55
284,    1,    7,    12,    0,    55
285,    0,    7,    12,    30.273,    55
286,    0,    7,    0,    29.455,    55
287,    0,    7,    0,    28.636,    55
288,    0,    8,    0,    27.818,    55
289,    0,    10,    0,    27,    55
290,    0,    11,    0,    26.182,    55
291,    0,    11,    0,    25.364,    55
292,    0,    11,    0,    24.545,    55
293,    0,    11,    0,    23.727,    55
294,    0,    11,    0,    22.909,    55
295,    0,    11,    0,    22.091,    55
296,    0,    11,    0,    21.273,    55
```

```
297,    0,    11,    0,    20.455,    55
298,    0,    11,    0,    19.636,    55
299,    0,    12,    0,    18.818,    55
300,    0,    14,    0,    18,    55
301,    0,    15,    0,    17.182,    55
302,    0,    15,    0,    16.364,    55
303,    0,    15,    0,    15.545,    55
304,    0,    15,    0,    14.727,    55
305,    0,    15,    0,    13.909,    55
306,    0,    15,    0,    13.091,    55
307,    0,    15,    0,    12.273,    55
308,    0,    15,    0,    11.455,    55
309,    0,    15,    0,    10.636,    55
310,    0,    16,    0,    9.818,    55
311,    0,    18,    0,    9,    55
312,    0,    19,    0,    8.182,    55
313,    0,    19,    0,    7.364,    55
314,    0,    19,    0,    6.545,    55
315,    0,    19,    0,    5.727,    55
316,    0,    19,    0,    4.909,    55
317,    0,    19,    0,    4.091,    55
318,    0,    19,    0,    3.273,    55
319,    0,    19,    0,    2.455,    55
320,    0,    19,    0,    1.636,    55
321,    0,    20,    0,    0.818,    55
322,    0,    22,    0,    0,    55
323,    0,    23,    0,    0.818,    55
324,    0,    23,    0,    0,    55
325,    1,    23,    0,    30.273,    55
326,    1,    20,    0,    29.455,    55
327,    1,    17,    0,    28.636,    55
328,    1,    15,    0,    27.818,    55
329,    1,    14,    0,    27,    55
330,    1,    12,    0,    26.182,    55
331,    1,    9,    0,    25.364,    55
332,    1,    8,    1,    24.545,    55
333,    1,    8,    3,    23.727,    55
334,    1,    8,    4,    22.909,    55
335,    1,    8,    4,    22.091,    55
336,    1,    8,    4,    21.273,    55
337,    1,    8,    4,    20.455,    55
338,    1,    8,    4,    19.636,    55
339,    1,    8,    4,    18.818,    55
340,    1,    8,    4,    18,    55
341,    1,    8,    4,    17.182,    55
342,    1,    8,    4,    16.364,    55
343,    1,    8,    5,    15.545,    55
344,    1,    8,    7,    14.727,    55
345,    1,    8,    8,    13.909,    55
346,    1,    7,    8,    13.091,    55
347,    1,    5,    8,    12.273,    55
348,    1,    4,    8,    11.455,    55
349,    1,    4,    8,    10.636,    55
350,    1,    5,    8,    9.818,    55
351,    1,    7,    8,    9,    55
352,    1,    8,    8,    8.182,    55
353,    1,    8,    8,    7.364,    55
354,    1,    8,    9,    6.545,    55
355,    1,    8,    11,    5.727,    55
356,    1,    8,    12,    4.909,    55
357,    1,    7,    12,    4.091,    55
358,    1,    5,    12,    3.273,    55
359,    1,    4,    12,    2.455,    55
360,    1,    4,    12,    1.636,    55
361,    1,    5,    12,    0.818,    55
362,    1,    7,    12,    0,    55
```

```
363,    0,    7,    12,    30.273,    55
364,    0,    7,    0,    29.455,    55
365,    0,    8,    0,    28.636,    55
366,    0,    10,    0,    27.818,    55
367,    0,    11,    0,    27,    55
368,    0,    11,    0,    26.182,    55
369,    0,    11,    0,    25.364,    55
370,    0,    11,    0,    24.545,    55
371,    0,    11,    0,    23.727,    55
372,    0,    11,    0,    22.909,    55
373,    0,    11,    0,    22.091,    55
374,    0,    11,    0,    21.273,    55
375,    0,    11,    0,    20.455,    55
376,    0,    12,    0,    19.636,    55
377,    0,    14,    0,    18.818,    55
378,    0,    15,    0,    18,    55
379,    0,    15,    0,    17.182,    55
380,    0,    15,    0,    16.364,    55
381,    0,    15,    0,    15.545,    55
382,    0,    15,    0,    14.727,    55
383,    0,    15,    0,    13.909,    55
384,    0,    15,    0,    13.091,    55
385,    0,    15,    0,    12.273,    55
386,    0,    15,    0,    11.455,    55
387,    0,    16,    0,    10.636,    55
388,    0,    18,    0,    9.818,    55
389,    0,    19,    0,    9,    55
390,    0,    19,    0,    8.182,    55
391,    0,    19,    0,    7.364,    55
392,    0,    19,    0,    6.545,    55
393,    0,    19,    0,    5.727,    55
394,    0,    19,    0,    4.909,    55
395,    0,    19,    0,    4.091,    55
396,    0,    19,    0,    3.273,    55
397,    0,    19,    0,    2.455,    55
398,    0,    20,    0,    1.636,    55
399,    0,    22,    0,    0.818,    55
400,    0,    23,    0,    0,    55
401,    1,    23,    0,    30.273,    55
402,    1,    20,    0,    29.455,    55
403,    1,    17,    0,    28.636,    55
404,    1,    15,    0,    27.818,    55
405,    1,    14,    0,    27,    55
406,    1,    12,    0,    26.182,    55
407,    1,    9,    0,    25.364,    55
408,    1,    8,    0,    24.545,    55
409,    1,    8,    1,    23.727,    55
410,    1,    8,    3,    22.909,    55
411,    1,    8,    4,    22.091,    55
412,    1,    8,    4,    21.273,    55
413,    1,    8,    4,    20.455,    55
414,    1,    8,    4,    19.636,    55
415,    1,    9,    4,    18.818,    55
416,    1,    10,    4,    18,    55
417,    1,    9,    4,    17.182,    55
418,    1,    8,    4,    16.364,    55
419,    1,    8,    4,    15.545,    55
420,    1,    8,    5,    14.727,    55
421,    1,    8,    7,    13.909,    55
422,    1,    7,    8,    13.091,    55
423,    1,    5,    8,    12.273,    55
424,    1,    4,    8,    11.455,    55
425,    1,    4,    8,    10.636,    55
426,    1,    5,    8,    9.818,    55
427,    1,    7,    8,    9,    55
428,    1,    8,    8,    8.182,    55
```

```
    429,          1,          8,          8,          7.364,          55
    430,          1,          8,          8,          6.545,          55
    431,          1,          8,          9,          5.727,          55
    432,          1,          8,          11,          4.909,          55
    433,          1,          7,          12,          4.091,          55
    434,          1,          5,          12,          3.273,          55
    435,          1,          4,          12,          2.455,          55
    436,          1,          4,          12,          1.636,          55
    437,          1,          5,          12,          0.818,          55
    438,          1,          7,          12,          0,          55
    439,          0,          7,          12,          30.273,          55
    440,          0,          7,          0,          29.455,          55
    441,          0,          7,          0,          28.636,          55
    442,          0,          8,          0,          27.818,          55
    443,          0,          10,          0,          27,          55
    444,          0,          11,          0,          26.182,          55
    445,          0,          11,          0,          25.364,          55
    446,          0,          11,          0,          24.545,          55
    447,          0,          11,          0,          23.727,          55
    448,          0,          11,          0,          22.909,          55
    449,          0,          11,          0,          22.091,          55
    450,          0,          11,          0,          21.273,          55
    451,          0,          11,          0,          20.455,          55
    452,          0,          11,          0,          19.636,          55
    453,          0,          12,          0,          18.818,          55
    454,          0,          14,          0,          18,          55
    455,          0,          15,          0,          17.182,          55
    456,          0,          15,          0,          16.364,          55
    457,          0,          15,          0,          15.545,          55
    458,          0,          15,          0,          14.727,          55
    459,          0,          15,          0,          13.909,          55
    460,          0,          15,          0,          13.091,          55
    461,          0,          15,          0,          12.273,          55
    462,          0,          15,          0,          11.455,          55
    463,          0,          15,          0,          10.636,          55
    464,          0,          16,          0,          9.818,          55
    465,          0,          18,          0,          9,          55
    466,          0,          19,          0,          8.182,          55
    467,          0,          19,          0,          7.364,          55
    468,          0,          19,          0,          6.545,          55
    469,          0,          19,          0,          5.727,          55
    470,          0,          19,          0,          4.909,          55
    471,          0,          19,          0,          4.091,          55
    472,          0,          19,          0,          3.273,          55
    473,          0,          19,          0,          2.455,          55
    474,          0,          19,          0,          1.636,          55
    475,          0,          20,          0,          0.818,          55
    476,          0,          22,          0,          0,          55
    477,          1,          23,          0,          30.273,          55
    478,          1,          20,          0,          29.455,          55
    479,          1,          17,          0,          28.636,          55
    480,          1,          15,          0,          27.818,          55
    481,          1,          14,          0,          27,          55
    482,          1,          12,          0,          26.182,          55
    483,          1,          9,          0,          25.364,          55
    484,          1,          8,          0,          24.545,          55
    485,          1,          8,          0,          23.727,          55
    486,          1,          8,          1,          22.909,          55
    487,          1,          8,          3,          22.091,          55
    488,          1,          8,          4,          21.273,          55
    489,          1,          8,          4,          20.455,          55
    490,          1,          8,          4,          19.636,          55
    491,          1,          9,          4,          18.818,          55
    492,          1,          11,          4,          18,          55
    493,          1,          11,          4,          17.182,          55
    494,          1,          9,          4,          16.364,          55
```

```
495,    1,    8,    4,    15.545,    55
496,    1,    8,    4,    14.727,    55
497,    1,    8,    5,    13.909,    55
498,    1,    7,    7,    13.091,    55
499,    1,    5,    8,    12.273,    55
500,    1,    4,    8,    11.455,    55
501,    1,    4,    8,    10.636,    55
502,    1,    5,    8,    9.818,    55
503,    1,    7,    8,    9,    55
504,    1,    8,    8,    8.182,    55
505,    1,    8,    8,    7.364,    55
506,    1,    8,    8,    6.545,    55
507,    1,    8,    8,    5.727,    55
508,    1,    8,    9,    4.909,    55
509,    1,    7,    11,    4.091,    55
510,    1,    5,    12,    3.273,    55
511,    1,    4,    12,    2.455,    55
512,    1,    4,    12,    1.636,    55
513,    1,    5,    12,    0.818,    55
514,    1,    7,    12,    0,    55
515,    0,    7,    12,    30.273,    55
516,    0,    7,    0,    29.455,    55
517,    0,    7,    0,    28.636,    55
518,    0,    7,    0,    27.818,    55
519,    0,    8,    0,    27,    55
520,    0,    10,    0,    26.182,    55
521,    0,    11,    0,    25.364,    55
522,    0,    11,    0,    24.545,    55
523,    0,    11,    0,    23.727,    55
524,    0,    11,    0,    22.909,    55
525,    0,    11,    0,    22.091,    55
526,    0,    11,    0,    21.273,    55
527,    0,    11,    0,    20.455,    55
528,    0,    11,    0,    19.636,    55
529,    0,    11,    0,    18.818,    55
530,    0,    12,    0,    18,    55
531,    0,    14,    0,    17.182,    55
532,    0,    15,    0,    16.364,    55
533,    0,    15,    0,    15.545,    55
534,    0,    15,    0,    14.727,    55
535,    0,    15,    0,    13.909,    55
536,    0,    15,    0,    13.091,    55
537,    0,    15,    0,    12.273,    55
538,    0,    15,    0,    11.455,    55
539,    0,    15,    0,    10.636,    55
540,    0,    15,    0,    9.818,    55
541,    0,    16,    0,    9,    55
542,    0,    18,    0,    8.182,    55
543,    0,    19,    0,    7.364,    55
544,    0,    19,    0,    6.545,    55
545,    0,    19,    0,    5.727,    55
546,    0,    19,    0,    4.909,    55
547,    0,    19,    0,    4.091,    55
548,    0,    19,    0,    3.273,    55
549,    0,    19,    0,    2.455,    55
550,    0,    19,    0,    1.636,    55
551,    0,    19,    0,    0.818,    55
552,    0,    20,    0,    0,    55
553,    1,    22,    0,    30.273,    55
554,    1,    19,    1,    29.455,    55
555,    1,    16,    1,    28.636,    55
556,    1,    14,    1,    27.818,    55
557,    1,    13,    1,    27,    55
558,    1,    11,    1,    26.182,    55
559,    1,    10,    1,    25.364,    55
560,    1,    8,    1,    24.545,    55
```

```
561,  1,  7,  1,  23.727,  55
562,  1,  7,  1,  22.909,  55
563,  1,  7,  2,  22.091,  55
564,  1,  7,  4,  21.273,  55
565,  1,  7,  5,  20.455,  55
566,  1,  7,  5,  19.636,  55
567,  1,  8,  5,  18.818,  55
568,  1,  10,  5,  18,  55
569,  1,  11,  5,  17.182,  55
570,  1,  10,  5,  16.364,  55
571,  1,  8,  5,  15.545,  55
572,  1,  8,  5,  14.727,  55
573,  1,  8,  5,  13.909,  55
574,  1,  7,  6,  13.091,  55
575,  1,  5,  8,  12.273,  55
576,  1,  4,  9,  11.455,  55
577,  1,  4,  9,  10.636,  55
578,  1,  5,  9,  9.818,  55
579,  1,  7,  9,  9,  55
580,  1,  8,  9,  8.182,  55
581,  1,  8,  9,  7.364,  55
582,  1,  8,  9,  6.545,  55
583,  1,  8,  9,  5.727,  55
584,  1,  8,  9,  4.909,  55
585,  1,  7,  10,  4.091,  55
586,  1,  5,  12,  3.273,  55
587,  1,  4,  13,  2.455,  55
588,  1,  4,  13,  1.636,  55
589,  1,  5,  13,  0.818,  55
590,  1,  7,  13,  0,  55
591,  0,  7,  13,  30.273,  55
592,  0,  7,  0,  29.455,  55
593,  0,  7,  0,  28.636,  55
594,  0,  7,  0,  27.818,  55
595,  0,  7,  0,  27,  55
596,  0,  8,  0,  26.182,  55
597,  0,  10,  0,  25.364,  55
598,  0,  11,  0,  24.545,  55
599,  0,  11,  0,  23.727,  55
600,  0,  11,  0,  22.909,  55
601,  0,  11,  0,  22.091,  55
602,  0,  11,  0,  21.273,  55
603,  0,  11,  0,  20.455,  55
604,  0,  11,  0,  19.636,  55
605,  0,  11,  0,  18.818,  55
606,  0,  11,  0,  18,  55
607,  0,  12,  0,  17.182,  55
608,  0,  14,  0,  16.364,  55
609,  0,  15,  0,  15.545,  55
610,  0,  15,  0,  14.727,  55
611,  0,  15,  0,  13.909,  55
612,  0,  15,  0,  13.091,  55
613,  0,  15,  0,  12.273,  55
614,  0,  15,  0,  11.455,  55
615,  0,  15,  0,  10.636,  55
616,  0,  15,  0,  9.818,  55
617,  0,  15,  0,  9,  55
618,  0,  16,  0,  8.182,  55
619,  0,  18,  0,  7.364,  55
620,  0,  19,  0,  6.545,  55
621,  0,  19,  0,  5.727,  55
622,  0,  19,  0,  4.909,  55
623,  0,  19,  0,  4.091,  55
624,  0,  19,  0,  3.273,  55
625,  0,  19,  0,  2.455,  55
626,  0,  19,  0,  1.636,  55
```

```
627,    0,    19,    0,    0.818,    55
628,    0,    19,    0,    0,    55
629,    1,    20,    0,    30.273,    55
630,    1,    17,    2,    29.455,    55
631,    1,    14,    3,    28.636,    55
632,    1,    12,    3,    27.818,    55
633,    1,    11,    3,    27,    55
634,    1,    9,    3,    26.182,    55
635,    1,    9,    3,    25.364,    55
636,    1,    8,    3,    24.545,    55
637,    1,    6,    3,    23.727,    55
638,    1,    5,    3,    22.909,    55
639,    1,    5,    3,    22.091,    55
640,    1,    5,    4,    21.273,    55
641,    1,    5,    6,    20.455,    55
642,    1,    5,    7,    19.636,    55
643,    1,    6,    7,    18.818,    55
644,    1,    8,    7,    18,    55
645,    1,    9,    7,    17.182,    55
646,    1,    9,    7,    16.364,    55
647,    1,    8,    7,    15.545,    55
648,    1,    8,    7,    14.727,    55
649,    1,    8,    7,    13.909,    55
650,    1,    7,    7,    13.091,    55
651,    1,    5,    8,    12.273,    55
652,    1,    4,    10,    11.455,    55
653,    1,    4,    11,    10.636,    55
654,    1,    5,    11,    9.818,    55
655,    1,    7,    11,    9,    55
656,    1,    8,    11,    8.182,    55
657,    1,    8,    11,    7.364,    55
658,    1,    8,    11,    6.545,    55
659,    1,    8,    11,    5.727,    55
660,    1,    8,    11,    4.909,    55
661,    1,    7,    11,    4.091,    55
662,    1,    5,    12,    3.273,    55
663,    1,    4,    14,    2.455,    55
664,    1,    4,    15,    1.636,    55
665,    1,    5,    15,    0.818,    55
666,    1,    7,    15,    0,    55
667,    0,    7,    15,    30.273,    55
668,    0,    7,    0,    29.455,    55
669,    0,    7,    0,    28.636,    55
670,    0,    7,    0,    27.818,    55
671,    0,    7,    0,    27,    55
672,    0,    7,    0,    26.182,    55
673,    0,    8,    0,    25.364,    55
674,    0,    10,    0,    24.545,    55
675,    0,    11,    0,    23.727,    55
676,    0,    11,    0,    22.909,    55
677,    0,    11,    0,    22.091,    55
678,    0,    11,    0,    21.273,    55
679,    0,    11,    0,    20.455,    55
680,    0,    11,    0,    19.636,    55
681,    0,    11,    0,    18.818,    55
682,    0,    11,    0,    18,    55
683,    0,    11,    0,    17.182,    55
684,    0,    12,    0,    16.364,    55
685,    0,    14,    0,    15.545,    55
686,    0,    15,    0,    14.727,    55
687,    0,    15,    0,    13.909,    55
688,    0,    15,    0,    13.091,    55
689,    0,    15,    0,    12.273,    55
690,    0,    15,    0,    11.455,    55
691,    0,    15,    0,    10.636,    55
692,    0,    15,    0,    9.818,    55
```

```
693,    0,    15,    0,    9,    55
694,    0,    15,    0,    8.182,    55
695,    0,    16,    0,    7.364,    55
696,    0,    18,    0,    6.545,    55
697,    0,    19,    0,    5.727,    55
698,    0,    19,    0,    4.909,    55
699,    0,    19,    0,    4.091,    55
700,    0,    19,    0,    3.273,    55
701,    0,    19,    0,    2.455,    55
702,    0,    19,    0,    1.636,    55
703,    0,    19,    0,    0.818,    55
704,    0,    19,    0,    0,    55
705,    1,    19,    0,    30.273,    55
706,    1,    16,    1,    29.455,    55
707,    1,    13,    3,    28.636,    55
708,    1,    11,    4,    27.818,    55
709,    1,    10,    4,    27,    55
710,    1,    8,    4,    26.182,    55
711,    1,    8,    4,    25.364,    55
712,    1,    8,    4,    24.545,    55
713,    1,    7,    4,    23.727,    55
714,    1,    5,    4,    22.909,    55
715,    1,    4,    4,    22.091,    55
716,    1,    4,    4,    21.273,    55
717,    1,    4,    5,    20.455,    55
718,    1,    4,    7,    19.636,    55
719,    1,    5,    8,    18.818,    55
720,    1,    7,    8,    18,    55
721,    1,    8,    8,    17.182,    55
722,    1,    8,    8,    16.364,    55
723,    1,    8,    8,    15.545,    55
724,    1,    8,    8,    14.727,    55
725,    1,    8,    8,    13.909,    55
726,    1,    7,    8,    13.091,    55
727,    1,    5,    8,    12.273,    55
728,    1,    4,    9,    11.455,    55
729,    1,    4,    11,    10.636,    55
730,    1,    5,    12,    9.818,    55
731,    1,    7,    12,    9,    55
732,    1,    8,    12,    8.182,    55
733,    1,    8,    12,    7.364,    55
734,    1,    8,    12,    6.545,    55
735,    1,    8,    12,    5.727,    55
736,    1,    8,    12,    4.909,    55
737,    1,    7,    12,    4.091,    55
738,    1,    5,    12,    3.273,    55
739,    1,    4,    13,    2.455,    55
740,    1,    4,    15,    1.636,    55
741,    1,    5,    16,    0.818,    55
742,    1,    7,    16,    0,    55
743,    0,    7,    16,    30.273,    55
744,    0,    7,    0,    29.455,    55
745,    0,    7,    0,    28.636,    55
746,    0,    7,    0,    27.818,    55
747,    0,    7,    0,    27,    55
748,    0,    7,    0,    26.182,    55
749,    0,    7,    0,    25.364,    55
750,    0,    8,    0,    24.545,    55
751,    0,    10,    0,    23.727,    55
752,    0,    11,    0,    22.909,    55
753,    0,    11,    0,    22.091,    55
754,    0,    11,    0,    21.273,    55
755,    0,    11,    0,    20.455,    55
756,    0,    11,    0,    19.636,    55
757,    0,    11,    0,    18.818,    55
758,    0,    11,    0,    18,    55
```

```
759,    0,    11,    0,    17.182,    55
760,    0,    11,    0,    16.364,    55
761,    0,    12,    0,    15.545,    55
762,    0,    14,    0,    14.727,    55
763,    0,    15,    0,    13.909,    55
764,    0,    15,    0,    13.091,    55
765,    0,    15,    0,    12.273,    55
766,    0,    15,    0,    11.455,    55
767,    0,    15,    0,    10.636,    55
768,    0,    15,    0,    9.818,    55
769,    0,    15,    0,    9,    55
770,    0,    15,    0,    8.182,    55
771,    0,    15,    0,    7.364,    55
772,    0,    16,    0,    6.545,    55
773,    0,    18,    0,    5.727,    55
774,    0,    19,    0,    4.909,    55
775,    0,    19,    0,    4.091,    55
776,    0,    19,    0,    3.273,    55
777,    0,    19,    0,    2.455,    55
778,    0,    19,    0,    1.636,    55
779,    0,    19,    0,    0.818,    55
780,    0,    19,    0,    0,    55
781,    1,    19,    0,    30.273,    55
782,    1,    16,    0,    29.455,    55
783,    1,    13,    1,    28.636,    55
784,    1,    11,    3,    27.818,    55
785,    1,    10,    4,    27,    55
786,    1,    8,    4,    26.182,    55
787,    1,    8,    4,    25.364,    55
788,    1,    8,    4,    24.545,    55
789,    1,    8,    4,    23.727,    55
790,    1,    7,    4,    22.909,    55
791,    1,    5,    4,    22.091,    55
792,    1,    4,    4,    21.273,    55
793,    1,    4,    4,    20.455,    55
794,    1,    4,    5,    19.636,    55
795,    1,    5,    7,    18.818,    55
796,    1,    7,    8,    18,    55
797,    1,    8,    8,    17.182,    55
798,    1,    8,    8,    16.364,    55
799,    1,    8,    8,    15.545,    55
800,    1,    8,    8,    14.727,    55
801,    1,    8,    8,    13.909,    55
802,    1,    7,    8,    13.091,    55
803,    1,    5,    8,    12.273,    55
804,    1,    4,    8,    11.455,    55
805,    1,    4,    9,    10.636,    55
806,    1,    5,    11,    9.818,    55
807,    1,    7,    12,    9,    55
808,    1,    8,    12,    8.182,    55
809,    1,    8,    12,    7.364,    55
810,    1,    8,    12,    6.545,    55
811,    1,    8,    12,    5.727,    55
812,    1,    8,    12,    4.909,    55
813,    1,    7,    12,    4.091,    55
814,    1,    5,    12,    3.273,    55
815,    1,    4,    12,    2.455,    55
816,    1,    4,    13,    1.636,    55
817,    1,    5,    15,    0.818,    55
818,    1,    7,    16,    0,    55
819,    0,    7,    16,    30.273,    55
820,    0,    7,    0,    29.455,    55
821,    0,    7,    0,    28.636,    55
822,    0,    7,    0,    27.818,    55
823,    0,    7,    0,    27,    55
824,    0,    7,    0,    26.182,    55
```

```
825,    0,    7,    0,    25.364,    55
826,    0,    7,    0,    24.545,    55
827,    0,    8,    0,    23.727,    55
828,    0,    10,    0,    22.909,    55
829,    0,    11,    0,    22.091,    55
830,    0,    11,    0,    21.273,    55
831,    0,    11,    0,    20.455,    55
832,    0,    11,    0,    19.636,    55
833,    0,    11,    0,    18.818,    55
834,    0,    11,    0,    18,    55
835,    0,    11,    0,    17.182,    55
836,    0,    11,    0,    16.364,    55
837,    0,    11,    0,    15.545,    55
838,    0,    12,    0,    14.727,    55
839,    0,    14,    0,    13.909,    55
840,    0,    15,    0,    13.091,    55
841,    0,    15,    0,    12.273,    55
842,    0,    15,    0,    11.455,    55
843,    0,    15,    0,    10.636,    55
844,    0,    15,    0,    9.818,    55
845,    0,    15,    0,    9,    55
846,    0,    15,    0,    8.182,    55
847,    0,    15,    0,    7.364,    55
848,    0,    15,    0,    6.545,    55
849,    0,    16,    0,    5.727,    55
850,    0,    18,    0,    4.909,    55
851,    0,    19,    0,    4.091,    55
852,    0,    19,    0,    3.273,    55
853,    0,    19,    0,    2.455,    55
854,    0,    19,    0,    1.636,    55
855,    0,    19,    0,    0.818,    55
856,    0,    19,    0,    0,    55
857,    0,    19,    0,    0.818,    55
858,    0,    19,    0,    0,    55
859,    1,    19,    0,    30.273,    55
860,    1,    16,    1,    29.455,    55
861,    1,    13,    3,    28.636,    55
862,    1,    11,    4,    27.818,    55
863,    1,    10,    4,    27,    55
864,    1,    8,    4,    26.182,    55
865,    1,    8,    4,    25.364,    55
866,    1,    8,    4,    24.545,    55
867,    1,    7,    4,    23.727,    55
868,    1,    5,    4,    22.909,    55
869,    1,    4,    4,    22.091,    55
870,    1,    4,    4,    21.273,    55
871,    1,    4,    5,    20.455,    55
872,    1,    4,    7,    19.636,    55
873,    1,    5,    8,    18.818,    55
874,    1,    7,    8,    18,    55
875,    1,    8,    8,    17.182,    55
876,    1,    8,    8,    16.364,    55
877,    1,    8,    8,    15.545,    55
878,    1,    8,    8,    14.727,    55
879,    1,    8,    8,    13.909,    55
880,    1,    7,    8,    13.091,    55
881,    1,    5,    8,    12.273,    55
882,    1,    4,    9,    11.455,    55
883,    1,    4,    11,    10.636,    55
884,    1,    5,    12,    9.818,    55
885,    1,    7,    12,    9,    55
886,    1,    8,    12,    8.182,    55
887,    1,    8,    12,    7.364,    55
888,    1,    8,    12,    6.545,    55
889,    1,    8,    12,    5.727,    55
890,    1,    8,    12,    4.909,    55
```

```
 891,  1,  7,  12,  4.091,  55
 892,  1,  5,  12,  3.273,  55
 893,  1,  4,  13,  2.455,  55
 894,  1,  4,  15,  1.636,  55
 895,  1,  5,  16,  0.818,  55
 896,  1,  7,  16,  0,  55
 897,  0,  7,  16,  30.273,  55
 898,  0,  7,  0,  29.455,  55
 899,  0,  7,  0,  28.636,  55
 900,  0,  7,  0,  27.818,  55
 901,  0,  7,  0,  27,  55
 902,  0,  7,  0,  26.182,  55
 903,  0,  7,  0,  25.364,  55
 904,  0,  8,  0,  24.545,  55
 905,  0,  10,  0,  23.727,  55
 906,  0,  11,  0,  22.909,  55
 907,  0,  11,  0,  22.091,  55
 908,  0,  11,  0,  21.273,  55
 909,  0,  11,  0,  20.455,  55
 910,  0,  11,  0,  19.636,  55
 911,  0,  11,  0,  18.818,  55
 912,  0,  11,  0,  18,  55
 913,  0,  11,  0,  17.182,  55
 914,  0,  11,  0,  16.364,  55
 915,  0,  12,  0,  15.545,  55
 916,  0,  14,  0,  14.727,  55
 917,  0,  15,  0,  13.909,  55
 918,  0,  15,  0,  13.091,  55
 919,  0,  15,  0,  12.273,  55
 920,  0,  15,  0,  11.455,  55
 921,  0,  15,  0,  10.636,  55
 922,  0,  15,  0,  9.818,  55
 923,  0,  15,  0,  9,  55
 924,  0,  15,  0,  8.182,  55
 925,  0,  15,  0,  7.364,  55
 926,  0,  16,  0,  6.545,  55
 927,  0,  18,  0,  5.727,  55
 928,  0,  19,  0,  4.909,  55
 929,  0,  19,  0,  4.091,  55
 930,  0,  19,  0,  3.273,  55
 931,  0,  19,  0,  2.455,  55
 932,  0,  19,  0,  1.636,  55
 933,  0,  19,  0,  0.818,  55
 934,  0,  19,  0,  0,  55
 935,  1,  19,  0,  30.273,  55
 936,  1,  16,  0,  29.455,  55
 937,  1,  13,  1,  28.636,  55
 938,  1,  11,  3,  27.818,  55
 939,  1,  10,  4,  27,  55
 940,  1,  8,  4,  26.182,  55
 941,  1,  8,  4,  25.364,  55
 942,  1,  8,  4,  24.545,  55
 943,  1,  8,  4,  23.727,  55
 944,  1,  7,  4,  22.909,  55
 945,  1,  5,  4,  22.091,  55
 946,  1,  4,  4,  21.273,  55
 947,  1,  4,  4,  20.455,  55
 948,  1,  4,  5,  19.636,  55
 949,  1,  5,  7,  18.818,  55
 950,  1,  7,  8,  18,  55
 951,  1,  8,  8,  17.182,  55
 952,  1,  8,  8,  16.364,  55
 953,  1,  8,  8,  15.545,  55
 954,  1,  8,  8,  14.727,  55
 955,  1,  8,  8,  13.909,  55
 956,  1,  7,  8,  13.091,  55
```

```
957,    1,    5,    8,    12.273,    55
958,    1,    4,    8,    11.455,    55
959,    1,    4,    9,    10.636,    55
960,    1,    5,    11,    9.818,    55
961,    1,    7,    12,    9,    55
962,    1,    8,    12,    8.182,    55
963,    1,    8,    12,    7.364,    55
964,    1,    8,    12,    6.545,    55
965,    1,    8,    12,    5.727,    55
966,    1,    8,    12,    4.909,    55
967,    1,    7,    12,    4.091,    55
968,    1,    5,    12,    3.273,    55
969,    1,    4,    12,    2.455,    55
970,    1,    4,    13,    1.636,    55
971,    1,    5,    15,    0.818,    55
972,    1,    7,    16,    0,    55
973,    0,    7,    16,    30.273,    55
974,    0,    7,    0,    29.455,    55
975,    0,    7,    0,    28.636,    55
976,    0,    7,    0,    27.818,    55
977,    0,    7,    0,    27,    55
978,    0,    7,    0,    26.182,    55
979,    0,    7,    0,    25.364,    55
980,    0,    7,    0,    24.545,    55
981,    0,    8,    0,    23.727,    55
982,    0,    10,    0,    22.909,    55
983,    0,    11,    0,    22.091,    55
984,    0,    11,    0,    21.273,    55
985,    0,    11,    0,    20.455,    55
986,    0,    11,    0,    19.636,    55
987,    0,    11,    0,    18.818,    55
988,    0,    11,    0,    18,    55
989,    0,    11,    0,    17.182,    55
990,    0,    11,    0,    16.364,    55
991,    0,    11,    0,    15.545,    55
992,    0,    12,    0,    14.727,    55
993,    0,    14,    0,    13.909,    55
994,    0,    15,    0,    13.091,    55
995,    0,    15,    0,    12.273,    55
996,    0,    15,    0,    11.455,    55
997,    0,    15,    0,    10.636,    55
998,    0,    15,    0,    9.818,    55
999,    0,    15,    0,    9,    55
1000,    0,    15,    0,    8.182,    55
1001,    0,    15,    0,    7.364,    55
1002,    0,    15,    0,    6.545,    55
1003,    0,    16,    0,    5.727,    55
1004,    0,    18,    0,    4.909,    55
1005,    0,    19,    0,    4.091,    55
1006,    0,    19,    0,    3.273,    55
1007,    0,    19,    0,    2.455,    55
1008,    0,    19,    0,    1.636,    55
1009,    0,    19,    0,    0.818,    55
1010,    0,    19,    0,    0,    55
1011,    1,    19,    0,    30.273,    55
1012,    1,    16,    0,    29.455,    55
1013,    1,    13,    0,    28.636,    55
1014,    1,    11,    1,    27.818,    55
1015,    1,    10,    3,    27,    55
1016,    1,    8,    4,    26.182,    55
1017,    1,    8,    4,    25.364,    55
1018,    1,    8,    4,    24.545,    55
1019,    1,    8,    4,    23.727,    55
1020,    1,    8,    4,    22.909,    55
1021,    1,    7,    4,    22.091,    55
1022,    1,    5,    4,    21.273,    55
```

```
1023,    1,    4,    4,    20.455,    55
1024,    1,    4,    4,    19.636,    55
1025,    1,    5,    5,    18.818,    55
1026,    1,    7,    7,    18,    55
1027,    1,    8,    8,    17.182,    55
1028,    1,    8,    8,    16.364,    55
1029,    1,    8,    8,    15.545,    55
1030,    1,    8,    8,    14.727,    55
1031,    1,    8,    8,    13.909,    55
1032,    1,    7,    8,    13.091,    55
1033,    1,    5,    8,    12.273,    55
1034,    1,    4,    8,    11.455,    55
1035,    1,    4,    8,    10.636,    55
1036,    1,    5,    9,    9.818,    55
1037,    1,    7,    11,    9,    55
1038,    1,    8,    12,    8.182,    55
1039,    1,    8,    12,    7.364,    55
1040,    1,    8,    12,    6.545,    55
1041,    1,    8,    12,    5.727,    55
1042,    1,    8,    12,    4.909,    55
1043,    1,    7,    12,    4.091,    55
1044,    1,    5,    12,    3.273,    55
1045,    1,    4,    12,    2.455,    55
1046,    1,    4,    12,    1.636,    55
1047,    1,    5,    13,    0.818,    55
1048,    1,    7,    15,    0,    55
1049,    0,    8,    16,    30.273,    55
1050,    0,    8,    0,    29.455,    55
1051,    0,    8,    0,    28.636,    55
1052,    0,    8,    0,    27.818,    55
1053,    0,    8,    0,    27,    55
1054,    0,    8,    0,    26.182,    55
1055,    0,    8,    0,    25.364,    55
1056,    0,    8,    0,    24.545,    55
1057,    0,    8,    0,    23.727,    55
1058,    0,    9,    0,    22.909,    55
1059,    0,    11,    0,    22.091,    55
1060,    0,    12,    0,    21.273,    55
1061,    0,    12,    0,    20.455,    55
1062,    0,    12,    0,    19.636,    55
1063,    0,    12,    0,    18.818,    55
1064,    0,    12,    0,    18,    55
1065,    0,    12,    0,    17.182,    55
1066,    0,    12,    0,    16.364,    55
1067,    0,    12,    0,    15.545,    55
1068,    0,    12,    0,    14.727,    55
1069,    0,    13,    0,    13.909,    55
1070,    0,    15,    0,    13.091,    55
1071,    0,    16,    0,    12.273,    55
1072,    0,    16,    0,    11.455,    55
1073,    0,    16,    0,    10.636,    55
1074,    0,    16,    0,    9.818,    55
1075,    0,    16,    0,    9,    55
1076,    0,    16,    0,    8.182,    55
1077,    0,    16,    0,    7.364,    55
1078,    0,    16,    0,    6.545,    55
1079,    0,    16,    0,    5.727,    55
1080,    0,    17,    0,    4.909,    55
1081,    0,    19,    0,    4.091,    55
1082,    0,    20,    0,    3.273,    55
1083,    0,    20,    0,    2.455,    55
1084,    0,    20,    0,    1.636,    55
1085,    0,    20,    0,    0.818,    55
1086,    0,    20,    0,    0,    55
1087,    1,    20,    0,    30.273,    55
1088,    1,    17,    0,    29.455,    55
```

```
     1089,              1,              14,              0,              28.636,              55
     1090,              1,              12,              0,              27.818,              55
     1091,              1,              11,              1,              27,              55
     1092,              1,              9,              3,              26.182,              55
     1093,              1,              8,              4,              25.364,              55
     1094,              1,              8,              4,              24.545,              55

"\\\\\ TRAMO: Tramo3 \\\\"
Iteracion,         Estado,       Vehículos,       Peatones,      Tiempo Restante,       Capacidades
     0,              0,              0,              0,              15.75,              2
     1,              0,              0,              0,              13.5,              2
     2,              0,              0,              0,              11.25,              2
     3,              0,              0,              0,              9,              2
     4,              0,              0,              0,              6.75,              2
     5,              0,              0,              0,              4.5,              2
     6,              0,              0,              0,              2.25,              2
     7,              0,              1,              0,              0,              2
     8,              1,              1,              0,              29.25,              2
     9,              1,              0,              0,              27,              2
     10,              1,              0,              0,              24.75,              2
     11,              1,              0,              2,              22.5,              2
     12,              1,              0,              4,              20.25,              2
     13,              1,              0,              6,              18,              2
     14,              1,              0,              7,              15.75,              2
     15,              1,              0,              8,              13.5,              2
     16,              1,              0,              9,              11.25,              2
     17,              1,              0,              10,              9,              2
     18,              1,              0,              11,              6.75,              2
     19,              1,              0,              11,              4.5,              2
     20,              1,              0,              11,              2.25,              2
     21,              1,              1,              11,              0,              2
     22,              0,              2,              13,              20.25,              2
     23,              0,              2,              0,              18,              2
     24,              0,              2,              0,              15.75,              2
     25,              0,              2,              0,              13.5,              2
     26,              0,              2,              0,              11.25,              2
     27,              0,              2,              0,              9,              2
     28,              0,              2,              0,              6.75,              2
     29,              0,              2,              0,              4.5,              2
     30,              0,              2,              0,              2.25,              2
     31,              0,              2,              0,              0,              2
     32,              1,              2,              0,              29.25,              2
     33,              1,              1,              2,              27,              2
     34,              1,              0,              4,              24.75,              2
     35,              1,              1,              6,              22.5,              2
     36,              1,              2,              7,              20.25,              2
     37,              1,              2,              8,              18,              2
     38,              1,              1,              9,              15.75,              2
     39,              1,              0,              10,              13.5,              2
     40,              1,              0,              11,              11.25,              2
     41,              1,              0,              11,              9,              2
     42,              1,              0,              11,              6.75,              2
     43,              1,              0,              11,              4.5,              2
     44,              1,              1,              13,              2.25,              2
     45,              1,              1,              15,              0,              2
     46,              0,              0,              17,              29.25,              2
     47,              0,              0,              0,              27,              2
     48,              0,              0,              0,              24.75,              2
     49,              0,              1,              0,              22.5,              2
     50,              0,              2,              0,              20.25,              2
     51,              0,              2,              0,              18,              2
     52,              0,              2,              0,              15.75,              2
     53,              0,              2,              0,              13.5,              2
     54,              0,              2,              0,              11.25,              2
     55,              0,              2,              0,              9,              2
     56,              0,              2,              0,              6.75,              2
```

```
57,  0,  2,  0,  4.5,  2
58,  0,  2,  0,  2.25,  2
59,  0,  2,  0,  0,  2
60,  1,  2,  0,  29.25,  2
61,  1,  1,  1,  27,  2
62,  1,  0,  2,  24.75,  2
63,  1,  1,  2,  22.5,  2
64,  1,  2,  2,  20.25,  2
65,  1,  1,  2,  18,  2
66,  1,  0,  4,  15.75,  2
67,  1,  0,  6,  13.5,  2
68,  1,  0,  8,  11.25,  2
69,  1,  0,  9,  9,  2
70,  1,  0,  10,  6.75,  2
71,  1,  0,  11,  4.5,  2
72,  1,  0,  12,  2.25,  2
73,  1,  0,  13,  0,  2
74,  0,  0,  13,  29.25,  2
75,  0,  0,  0,  27,  2
76,  0,  0,  0,  24.75,  2
77,  0,  1,  0,  22.5,  2
78,  0,  2,  0,  20.25,  2
79,  0,  2,  0,  18,  2
80,  0,  2,  0,  15.75,  2
81,  0,  2,  0,  13.5,  2
82,  0,  2,  0,  11.25,  2
83,  0,  2,  0,  9,  2
84,  0,  2,  0,  6.75,  2
85,  0,  2,  0,  4.5,  2
86,  0,  2,  0,  2.25,  2
87,  0,  2,  0,  0,  2
88,  1,  2,  0,  29.25,  2
89,  1,  1,  2,  27,  2
90,  1,  1,  4,  24.75,  2
91,  1,  2,  5,  22.5,  2
92,  1,  2,  6,  20.25,  2
93,  1,  1,  7,  18,  2
94,  1,  0,  8,  15.75,  2
95,  1,  0,  9,  13.5,  2
96,  1,  0,  9,  11.25,  2
97,  1,  0,  9,  9,  2
98,  1,  1,  9,  6.75,  2
99,  1,  1,  11,  4.5,  2
100,  1,  0,  13,  2.25,  2
101,  1,  0,  15,  0,  2
102,  0,  1,  16,  29.25,  2
103,  0,  1,  0,  27,  2
104,  0,  1,  0,  24.75,  2
105,  0,  2,  0,  22.5,  2
106,  0,  2,  0,  20.25,  2
107,  0,  2,  0,  18,  2
108,  0,  2,  0,  15.75,  2
109,  0,  2,  0,  13.5,  2
110,  0,  2,  0,  11.25,  2
111,  0,  2,  0,  9,  2
112,  0,  2,  0,  6.75,  2
113,  0,  2,  0,  4.5,  2
114,  0,  2,  0,  2.25,  2
115,  0,  2,  0,  0,  2
116,  1,  2,  0,  29.25,  2
117,  1,  1,  1,  27,  2
118,  1,  0,  1,  24.75,  2
119,  1,  1,  1,  22.5,  2
120,  1,  2,  1,  20.25,  2
121,  1,  1,  3,  18,  2
122,  1,  0,  5,  15.75,  2
```

```
123,    1,    0,    7,    13.5,    2
124,    1,    0,    8,    11.25,    2
125,    1,    0,    9,    9,    2
126,    1,    0,    10,    6.75,    2
127,    1,    0,    11,    4.5,    2
128,    1,    0,    12,    2.25,    2
129,    1,    0,    12,    0,    2
130,    0,    0,    12,    29.25,    2
131,    0,    0,    0,    27,    2
132,    0,    1,    0,    24.75,    2
133,    0,    2,    0,    22.5,    2
134,    0,    2,    0,    20.25,    2
135,    0,    2,    0,    18,    2
136,    0,    2,    0,    15.75,    2
137,    0,    2,    0,    13.5,    2
138,    0,    2,    0,    11.25,    2
139,    0,    2,    0,    9,    2
140,    0,    2,    0,    6.75,    2
141,    0,    2,    0,    4.5,    2
142,    0,    2,    0,    2.25,    2
143,    0,    2,    0,    0,    2
144,    1,    2,    0,    29.25,    2
145,    1,    1,    2,    27,    2
146,    1,    1,    3,    24.75,    2
147,    1,    2,    4,    22.5,    2
148,    1,    2,    5,    20.25,    2
149,    1,    1,    6,    18,    2
150,    1,    0,    7,    15.75,    2
151,    1,    0,    7,    13.5,    2
152,    1,    0,    7,    11.25,    2
153,    1,    0,    7,    9,    2
154,    1,    1,    9,    6.75,    2
155,    1,    1,    11,    4.5,    2
156,    1,    0,    13,    2.25,    2
157,    1,    0,    14,    0,    2
158,    0,    1,    15,    29.25,    2
159,    0,    1,    0,    27,    2
160,    0,    2,    0,    24.75,    2
161,    0,    3,    0,    22.5,    2
162,    0,    3,    0,    20.25,    2
163,    0,    3,    0,    18,    2
164,    0,    3,    0,    15.75,    2
165,    0,    3,    0,    13.5,    2
166,    0,    3,    0,    11.25,    2
167,    0,    3,    0,    9,    2
168,    0,    3,    0,    6.75,    2
169,    0,    3,    0,    4.5,    2
170,    0,    3,    0,    2.25,    2
171,    0,    3,    0,    0,    2
172,    1,    3,    0,    29.25,    2
173,    1,    2,    0,    27,    2
174,    1,    1,    0,    24.75,    2
175,    1,    2,    0,    22.5,    2
176,    1,    1,    2,    20.25,    2
177,    1,    0,    4,    18,    2
178,    1,    0,    6,    15.75,    2
179,    1,    0,    7,    13.5,    2
180,    1,    0,    8,    11.25,    2
181,    1,    0,    9,    9,    2
182,    1,    0,    10,    6.75,    2
183,    1,    0,    11,    4.5,    2
184,    1,    0,    11,    2.25,    2
185,    1,    0,    11,    0,    2
186,    0,    0,    11,    29.25,    2
187,    0,    0,    0,    27,    2
188,    0,    1,    0,    24.75,    2
```

```
189,    0,    2,    0,    22.5,    2
190,    0,    2,    0,    20.25,    2
191,    0,    2,    0,    18,    2
192,    0,    2,    0,    15.75,    2
193,    0,    2,    0,    13.5,    2
194,    0,    2,    0,    11.25,    2
195,    0,    2,    0,    9,    2
196,    0,    2,    0,    6.75,    2
197,    0,    2,    0,    4.5,    2
198,    0,    2,    0,    2.25,    2
199,    0,    2,    0,    0,    2
200,    1,    2,    0,    29.25,    2
201,    1,    1,    1,    27,    2
202,    1,    1,    2,    24.75,    2
203,    1,    2,    3,    22.5,    2
204,    1,    1,    4,    20.25,    2
205,    1,    0,    5,    18,    2
206,    1,    0,    5,    15.75,    2
207,    1,    0,    5,    13.5,    2
208,    1,    0,    5,    11.25,    2
209,    1,    0,    7,    9,    2
210,    1,    0,    9,    6.75,    2
211,    1,    0,    11,    4.5,    2
212,    1,    0,    12,    2.25,    2
213,    1,    0,    13,    0,    2
214,    0,    0,    14,    29.25,    2
215,    0,    0,    0,    27,    2
216,    0,    1,    0,    24.75,    2
217,    0,    2,    0,    22.5,    2
218,    0,    2,    0,    20.25,    2
219,    0,    2,    0,    18,    2
220,    0,    2,    0,    15.75,    2
221,    0,    2,    0,    13.5,    2
222,    0,    2,    0,    11.25,    2
223,    0,    2,    0,    9,    2
224,    0,    2,    0,    6.75,    2
225,    0,    2,    0,    4.5,    2
226,    0,    2,    0,    2.25,    2
227,    0,    2,    0,    0,    2
228,    1,    2,    0,    29.25,    2
229,    1,    2,    0,    27,    2
230,    1,    2,    0,    24.75,    2
231,    1,    1,    2,    22.5,    2
232,    1,    0,    4,    20.25,    2
233,    1,    0,    6,    18,    2
234,    1,    0,    7,    15.75,    2
235,    1,    0,    8,    13.5,    2
236,    1,    0,    9,    11.25,    2
237,    1,    1,    10,    9,    2
238,    1,    1,    11,    6.75,    2
239,    1,    0,    11,    4.5,    2
240,    1,    0,    11,    2.25,    2
241,    1,    1,    11,    0,    2
242,    0,    1,    13,    20.25,    2
243,    0,    2,    0,    18,    2
244,    0,    3,    0,    15.75,    2
245,    0,    3,    0,    13.5,    2
246,    0,    3,    0,    11.25,    2
247,    0,    3,    0,    9,    2
248,    0,    3,    0,    6.75,    2
249,    0,    3,    0,    4.5,    2
250,    0,    3,    0,    2.25,    2
251,    0,    3,    0,    0,    2
252,    1,    3,    0,    29.25,    2
253,    1,    2,    2,    27,    2
254,    1,    0,    4,    24.75,    2
```

```
255,    1,    0,    6,    22.5,    2
256,    1,    0,    7,    20.25,    2
257,    1,    1,    8,    18,    2
258,    1,    2,    9,    15.75,    2
259,    1,    1,    10,    13.5,    2
260,    1,    0,    11,    11.25,    2
261,    1,    0,    11,    9,    2
262,    1,    0,    11,    6.75,    2
263,    1,    0,    11,    4.5,    2
264,    1,    0,    13,    2.25,    2
265,    1,    0,    15,    0,    2
266,    0,    0,    17,    29.25,    2
267,    0,    0,    0,    27,    2
268,    0,    0,    0,    24.75,    2
269,    0,    0,    0,    22.5,    2
270,    0,    0,    0,    20.25,    2
271,    0,    1,    0,    18,    2
272,    0,    2,    0,    15.75,    2
273,    0,    2,    0,    13.5,    2
274,    0,    2,    0,    11.25,    2
275,    0,    2,    0,    9,    2
276,    0,    2,    0,    6.75,    2
277,    0,    2,    0,    4.5,    2
278,    0,    2,    0,    2.25,    2
279,    0,    2,    0,    0,    2
280,    1,    2,    0,    29.25,    2
281,    1,    1,    1,    27,    2
282,    1,    0,    2,    24.75,    2
283,    1,    0,    2,    22.5,    2
284,    1,    1,    2,    20.25,    2
285,    1,    2,    2,    18,    2
286,    1,    2,    4,    15.75,    2
287,    1,    1,    6,    13.5,    2
288,    1,    0,    8,    11.25,    2
289,    1,    0,    9,    9,    2
290,    1,    0,    10,    6.75,    2
291,    1,    0,    11,    4.5,    2
292,    1,    1,    12,    2.25,    2
293,    1,    1,    13,    0,    2
294,    0,    0,    13,    20.25,    2
295,    0,    0,    0,    18,    2
296,    0,    1,    0,    15.75,    2
297,    0,    1,    0,    13.5,    2
298,    0,    1,    0,    11.25,    2
299,    0,    2,    0,    9,    2
300,    0,    2,    0,    6.75,    2
301,    0,    2,    0,    4.5,    2
302,    0,    2,    0,    2.25,    2
303,    0,    2,    0,    0,    2
304,    1,    2,    0,    29.25,    2
305,    1,    1,    0,    27,    2
306,    1,    0,    0,    24.75,    2
307,    1,    0,    0,    22.5,    2
308,    1,    0,    2,    20.25,    2
309,    1,    0,    4,    18,    2
310,    1,    0,    6,    15.75,    2
311,    1,    0,    7,    13.5,    2
312,    1,    0,    8,    11.25,    2
313,    1,    1,    9,    9,    2
314,    1,    2,    10,    6.75,    2
315,    1,    1,    11,    4.5,    2
316,    1,    0,    11,    2.25,    2
317,    1,    0,    11,    0,    2
318,    0,    0,    11,    29.25,    2
319,    0,    0,    0,    27,    2
320,    0,    0,    0,    24.75,    2
```

```
321,    0,    0,    0,    22.5,    2
322,    0,    0,    0,    20.25,    2
323,    0,    0,    0,    18,    2
324,    0,    0,    0,    15.75,    2
325,    0,    0,    0,    13.5,    2
326,    0,    0,    0,    11.25,    2
327,    0,    1,    0,    9,    2
328,    0,    2,    0,    6.75,    2
329,    0,    2,    0,    4.5,    2
330,    0,    2,    0,    2.25,    2
331,    0,    2,    0,    0,    2
332,    1,    2,    0,    29.25,    2
333,    1,    1,    1,    27,    2
334,    1,    0,    2,    24.75,    2
335,    1,    0,    3,    22.5,    2
336,    1,    0,    4,    20.25,    2
337,    1,    0,    5,    18,    2
338,    1,    0,    5,    15.75,    2
339,    1,    0,    5,    13.5,    2
340,    1,    1,    5,    11.25,    2
341,    1,    2,    7,    9,    2
342,    1,    2,    9,    6.75,    2
343,    1,    1,    11,    4.5,    2
344,    1,    0,    12,    2.25,    2
345,    1,    0,    13,    0,    2
346,    0,    0,    14,    29.25,    2
347,    0,    0,    0,    27,    2
348,    0,    1,    0,    24.75,    2
349,    0,    1,    0,    22.5,    2
350,    0,    1,    0,    20.25,    2
351,    0,    1,    0,    18,    2
352,    0,    2,    0,    15.75,    2
353,    0,    2,    0,    13.5,    2
354,    0,    3,    0,    11.25,    2
355,    0,    4,    0,    9,    2
356,    0,    4,    0,    6.75,    2
357,    0,    4,    0,    4.5,    2
358,    0,    4,    0,    2.25,    2
359,    0,    4,    0,    0,    2
360,    1,    4,    0,    29.25,    2
361,    1,    3,    0,    27,    2
362,    1,    0,    0,    24.75,    2
363,    1,    0,    2,    22.5,    2
364,    1,    0,    4,    20.25,    2
365,    1,    0,    6,    18,    2
366,    1,    0,    7,    15.75,    2
367,    1,    0,    8,    13.5,    2
368,    1,    1,    9,    11.25,    2
369,    1,    2,    10,    9,    2
370,    1,    1,    11,    6.75,    2
371,    1,    0,    11,    4.5,    2
372,    1,    0,    11,    2.25,    2
373,    1,    0,    11,    0,    2
374,    0,    0,    13,    29.25,    2
375,    0,    0,    0,    27,    2
376,    0,    0,    0,    24.75,    2
377,    0,    0,    0,    22.5,    2
378,    0,    0,    0,    20.25,    2
379,    0,    0,    0,    18,    2
380,    0,    0,    0,    15.75,    2
381,    0,    0,    0,    13.5,    2
382,    0,    1,    0,    11.25,    2
383,    0,    2,    0,    9,    2
384,    0,    2,    0,    6.75,    2
385,    0,    2,    0,    4.5,    2
386,    0,    2,    0,    2.25,    2
```

```
     387,              0,              2,              0,              0,              2
     388,              1,              2,              0,             45,              2
     389,              1,              1,              1,          42.75,              2
     390,              1,              0,              2,           40.5,              2
     391,              1,              0,              3,          38.25,              2
     392,              1,              0,              4,             36,              2
     393,              1,              0,              4,          33.75,              2
     394,              1,              0,              4,           31.5,              2
     395,              1,              0,              4,          29.25,              2
     396,              1,              1,              6,             27,              2
     397,              1,              2,              8,          24.75,              2
     398,              1,              1,             10,           22.5,              2

"\\\\\ TRAMO: Tramo4 \\\\"
Iteracion,         Estado,       Vehículos,       Peatones,      Tiempo Restante,       Capacidades
     0,              0,              0,              0,             15,              6
     1,              0,              0,              0,           13.5,              6
     2,              0,              0,              0,             12,              6
     3,              0,              0,              0,           10.5,              6
     4,              0,              0,              0,              9,              6
     5,              0,              0,              0,            7.5,              6
     6,              0,              0,              0,              6,              6
     7,              0,              0,              0,            4.5,              6
     8,              0,              0,              0,              3,              6
     9,              0,              0,              0,            1.5,              6
     10,             0,              0,              0,              0,              6
     11,             1,              1,              0,             30,              6
     12,             1,              2,              7,           28.5,              6
     13,             1,              1,             13,             27,              6
     14,             1,              0,             19,           25.5,              6
     15,             1,              0,             24,             24,              6
     16,             1,              0,             28,           22.5,              6
     17,             1,              1,             31,             21,              6
     18,             1,              2,             33,           19.5,              6
     19,             1,              1,             35,             18,              6
     20,             1,              0,             36,           16.5,              6
     21,             1,              0,             36,             15,              6
     22,             1,              1,             44,           13.5,              6
     23,             1,              1,             51,             12,              6
     24,             1,              0,             57,           10.5,              6
     25,             1,              0,             63,              9,              6
     26,             1,              0,             68,            7.5,              6
     27,             1,              0,             72,              6,              6
     28,             1,              1,             75,            4.5,              6
     29,             1,              1,             77,              3,              6
     30,             1,              0,             79,            1.5,              6
     31,             1,              0,             80,              0,              6
     32,             0,              1,             80,             30,              6
     33,             0,              1,              0,           28.5,              6
     34,             0,              1,              0,             27,              6
     35,             0,              1,              0,           25.5,              6
     36,             0,              1,              0,             24,              6
     37,             0,              1,              0,           22.5,              6
     38,             0,              1,              0,             21,              6
     39,             0,              1,              0,           19.5,              6
     40,             0,              1,              0,             18,              6
     41,             0,              1,              0,           16.5,              6
     42,             0,              1,              0,             15,              6
     43,             0,              1,              0,           13.5,              6
     44,             0,              1,              0,             12,              6
     45,             0,              1,              0,           10.5,              6
     46,             0,              1,              0,              9,              6
     47,             0,              1,              0,            7.5,              6
     48,             0,              1,              0,              6,              6
     49,             0,              1,              0,            4.5,              6
     50,             0,              1,              0,              3,              6
```

```
51,   0,   1,   0,   1.5,   6
52,   0,   3,   0,   0,   6
53,   1,   5,   0,   30,   6
54,   1,   3,   0,   28.5,   6
55,   1,   3,   8,   27,   6
56,   1,   3,   15,   25.5,   6
57,   1,   1,   21,   24,   6
58,   1,   0,   27,   22.5,   6
59,   1,   1,   32,   21,   6
60,   1,   2,   36,   19.5,   6
61,   1,   1,   39,   18,   6
62,   1,   0,   41,   16.5,   6
63,   1,   0,   43,   15,   6
64,   1,   1,   44,   13.5,   6
65,   1,   1,   44,   12,   6
66,   1,   0,   52,   10.5,   6
67,   1,   1,   59,   9,   6
68,   1,   1,   65,   7.5,   6
69,   1,   0,   71,   6,   6
70,   1,   1,   76,   4.5,   6
71,   1,   1,   80,   3,   6
72,   1,   0,   83,   1.5,   6
73,   1,   0,   85,   0,   6
74,   0,   1,   87,   30,   6
75,   0,   1,   0,   28.5,   6
76,   0,   1,   0,   27,   6
77,   0,   1,   0,   25.5,   6
78,   0,   1,   0,   24,   6
79,   0,   1,   0,   22.5,   6
80,   0,   1,   0,   21,   6
81,   0,   1,   0,   19.5,   6
82,   0,   1,   0,   18,   6
83,   0,   1,   0,   16.5,   6
84,   0,   1,   0,   15,   6
85,   0,   1,   0,   13.5,   6
86,   0,   1,   0,   12,   6
87,   0,   1,   0,   10.5,   6
88,   0,   1,   0,   9,   6
89,   0,   1,   0,   7.5,   6
90,   0,   1,   0,   6,   6
91,   0,   1,   0,   4.5,   6
92,   0,   1,   0,   3,   6
93,   0,   1,   0,   1.5,   6
94,   0,   3,   0,   0,   6
95,   1,   5,   0,   30,   6
96,   1,   3,   2,   28.5,   6
97,   1,   3,   3,   27,   6
98,   1,   2,   3,   25.5,   6
99,   1,   0,   11,   24,   6
100,   1,   0,   18,   22.5,   6
101,   1,   1,   24,   21,   6
102,   1,   2,   30,   19.5,   6
103,   1,   1,   35,   18,   6
104,   1,   0,   39,   16.5,   6
105,   1,   1,   42,   15,   6
106,   1,   2,   44,   13.5,   6
107,   1,   1,   46,   12,   6
108,   1,   0,   47,   10.5,   6
109,   1,   0,   47,   9,   6
110,   1,   0,   55,   7.5,   6
111,   1,   1,   62,   6,   6
112,   1,   2,   68,   4.5,   6
113,   1,   1,   74,   3,   6
114,   1,   0,   79,   1.5,   6
115,   1,   0,   83,   0,   6
116,   0,   1,   86,   30,   6
```

```
    117,            0,            1,            0,           28.5,            6
    118,            0,            1,            0,             27,            6
    119,            0,            1,            0,           25.5,            6
    120,            0,            1,            0,             24,            6
    121,            0,            1,            0,           22.5,            6
    122,            0,            1,            0,             21,            6
    123,            0,            1,            0,           19.5,            6
    124,            0,            1,            0,             18,            6
    125,            0,            1,            0,           16.5,            6
    126,            0,            1,            0,             15,            6
    127,            0,            1,            0,           13.5,            6
    128,            0,            1,            0,             12,            6
    129,            0,            1,            0,           10.5,            6
    130,            0,            1,            0,              9,            6
    131,            0,            1,            0,            7.5,            6
    132,            0,            1,            0,              6,            6
    133,            0,            1,            0,            4.5,            6
    134,            0,            1,            0,              3,            6
    135,            0,            2,            0,            1.5,            6
    136,            0,            4,            0,              0,            6
    137,            1,            6,            0,             30,            6
    138,            1,            4,            3,           28.5,            6
    139,            1,            2,            5,             27,            6
    140,            1,            1,            7,           25.5,            6
    141,            1,            0,            8,             24,            6
    142,            1,            0,            8,           22.5,            6
    143,            1,            1,           16,             21,            6
    144,            1,            2,           23,           19.5,            6
    145,            1,            1,           29,             18,            6
    146,            1,            0,           35,           16.5,            6
    147,            1,            1,           40,             15,            6
    148,            1,            1,           44,           13.5,            6
    149,            1,            0,           47,             12,            6
    150,            1,            0,           49,           10.5,            6
    151,            1,            0,           51,              9,            6
    152,            1,            0,           52,            7.5,            6
    153,            1,            1,           52,              6,            6
    154,            1,            1,           60,            4.5,            6
    155,            1,            0,           67,              3,            6
    156,            1,            0,           73,            1.5,            6
    157,            1,            1,           79,              0,            6
    158,            0,            2,           84,             30,            6
    159,            0,            2,            0,           28.5,            6
    160,            0,            2,            0,             27,            6
    161,            0,            2,            0,           25.5,            6
    162,            0,            2,            0,             24,            6
    163,            0,            2,            0,           22.5,            6
    164,            0,            2,            0,             21,            6
    165,            0,            2,            0,           19.5,            6
    166,            0,            2,            0,             18,            6
    167,            0,            2,            0,           16.5,            6
    168,            0,            2,            0,             15,            6
    169,            0,            2,            0,           13.5,            6
    170,            0,            2,            0,             12,            6
    171,            0,            2,            0,           10.5,            6
    172,            0,            2,            0,              9,            6
    173,            0,            2,            0,            7.5,            6
    174,            0,            2,            0,              6,            6
    175,            0,            2,            0,            4.5,            6
    176,            0,            2,            0,              3,            6
    177,            0,            2,            0,            1.5,            6
    178,            0,            4,            0,              0,            6
    179,            1,            6,            0,             30,            6
    180,            1,            4,            5,           28.5,            6
    181,            1,            3,            9,             27,            6
    182,            1,            2,           12,           25.5,            6
```

```
183,    1,    0,    14,    24,    6
184,    1,    0,    16,    22.5,    6
185,    1,    1,    17,    21,    6
186,    1,    2,    17,    19.5,    6
187,    1,    1,    25,    18,    6
188,    1,    0,    32,    16.5,    6
189,    1,    1,    38,    15,    6
190,    1,    2,    44,    13.5,    6
191,    1,    1,    49,    12,    6
192,    1,    0,    53,    10.5,    6
193,    1,    0,    56,    9,    6
194,    1,    0,    58,    7.5,    6
195,    1,    1,    60,    6,    6
196,    1,    2,    61,    4.5,    6
197,    1,    1,    61,    3,    6
198,    1,    0,    69,    1.5,    6
199,    1,    1,    76,    0,    6
200,    0,    2,    82,    30,    6
201,    0,    2,    0,    28.5,    6
202,    0,    2,    0,    27,    6
203,    0,    2,    0,    25.5,    6
204,    0,    2,    0,    24,    6
205,    0,    2,    0,    22.5,    6
206,    0,    2,    0,    21,    6
207,    0,    2,    0,    19.5,    6
208,    0,    2,    0,    18,    6
209,    0,    2,    0,    16.5,    6
210,    0,    2,    0,    15,    6
211,    0,    2,    0,    13.5,    6
212,    0,    2,    0,    12,    6
213,    0,    2,    0,    10.5,    6
214,    0,    2,    0,    9,    6
215,    0,    2,    0,    7.5,    6
216,    0,    2,    0,    6,    6
217,    0,    2,    0,    4.5,    6
218,    0,    2,    0,    3,    6
219,    0,    3,    0,    1.5,    6
220,    0,    5,    0,    0,    6
221,    1,    7,    0,    30,    6
222,    1,    5,    6,    28.5,    6
223,    1,    2,    12,    27,    6
224,    1,    1,    17,    25.5,    6
225,    1,    0,    21,    24,    6
226,    1,    0,    24,    22.5,    6
227,    1,    1,    26,    21,    6
228,    1,    2,    28,    19.5,    6
229,    1,    1,    29,    18,    6
230,    1,    0,    29,    16.5,    6
231,    1,    1,    37,    15,    6
232,    1,    1,    44,    13.5,    6
233,    1,    0,    50,    12,    6
234,    1,    0,    56,    10.5,    6
235,    1,    0,    61,    9,    6
236,    1,    0,    65,    7.5,    6
237,    1,    1,    68,    6,    6
238,    1,    1,    70,    4.5,    6
239,    1,    0,    72,    3,    6
240,    1,    0,    73,    1.5,    6
241,    1,    1,    73,    0,    6
242,    0,    1,    81,    30,    6
243,    0,    1,    0,    28.5,    6
244,    0,    1,    0,    27,    6
245,    0,    1,    0,    25.5,    6
246,    0,    1,    0,    24,    6
247,    0,    1,    0,    22.5,    6
248,    0,    1,    0,    21,    6
```

```
    249,            0,            1,            0,          19.5,             6
    250,            0,            1,            0,            18,           6
    251,            0,            1,            0,          16.5,             6
    252,            0,            1,            0,            15,           6
    253,            0,            1,            0,          13.5,             6
    254,            0,            1,            0,            12,           6
    255,            0,            1,            0,          10.5,             6
    256,            0,            1,            0,             9,           6
    257,            0,            1,            0,           7.5,             6
    258,            0,            1,            0,             6,           6
    259,            0,            1,            0,           4.5,             6
    260,            0,            1,            0,             3,           6
    261,            0,            2,            0,           1.5,             6
    262,            0,            4,            0,             0,           6
    263,            1,            6,            0,            30,           6
    264,            1,            4,            8,          28.5,             6
    265,            1,            1,           15,            27,             6
    266,            1,            0,           21,          25.5,               6
    267,            1,            0,           27,            24,             6
    268,            1,            0,           32,          22.5,               6
    269,            1,            1,           36,            21,             6
    270,            1,            2,           39,          19.5,               6
    271,            1,            1,           41,            18,             6
    272,            1,            1,           43,          16.5,               6
    273,            1,            2,           44,            15,             6
    274,            1,            1,           44,          13.5,               6
    275,            1,            0,           52,            12,             6
    276,            1,            0,           59,          10.5,               6
    277,            1,            0,           65,             9,           6
    278,            1,            1,           71,           7.5,             6
    279,            1,            2,           76,             6,           6
    280,            1,            1,           80,           4.5,             6
    281,            1,            0,           83,             3,           6
    282,            1,            0,           85,           1.5,             6
    283,            1,            1,           87,             0,           6
    284,            0,            1,           88,            30,           6
    285,            0,            1,            0,          28.5,             6
    286,            0,            1,            0,            27,           6
    287,            0,            1,            0,          25.5,             6
    288,            0,            1,            0,            24,           6
    289,            0,            1,            0,          22.5,             6
    290,            0,            1,            0,            21,           6
    291,            0,            1,            0,          19.5,             6
    292,            0,            1,            0,            18,           6
    293,            0,            1,            0,          16.5,             6
    294,            0,            1,            0,            15,           6
    295,            0,            1,            0,          13.5,             6
    296,            0,            1,            0,            12,           6
    297,            0,            1,            0,          10.5,             6
    298,            0,            1,            0,             9,           6
    299,            0,            1,            0,           7.5,             6
    300,            0,            1,            0,             6,           6
    301,            0,            1,            0,           4.5,             6
    302,            0,            3,            0,             3,           6
    303,            0,            4,            0,           1.5,             6
    304,            0,            6,            0,             0,           6
    305,            1,            7,            0,            30,           6
    306,            1,            5,            1,          28.5,             6
    307,            1,            1,            1,            27,           6
    308,            1,            0,            9,          25.5,             6
    309,            1,            0,           16,            24,             6
    310,            1,            0,           22,          22.5,               6
    311,            1,            1,           28,            21,             6
    312,            1,            1,           33,          19.5,               6
    313,            1,            0,           37,            18,             6
    314,            1,            1,           40,          16.5,               6
```

```
315,    1,    1,    42,    15,    6
316,    1,    0,    44,    13.5,    6
317,    1,    0,    45,    12,    6
318,    1,    0,    45,    10.5,    6
319,    1,    0,    53,    9,    6
320,    1,    1,    60,    7.5,    6
321,    1,    1,    66,    6,    6
322,    1,    0,    72,    4.5,    6
323,    1,    1,    77,    3,    6
324,    1,    1,    81,    1.5,    6
325,    1,    1,    84,    0,    6
326,    0,    1,    86,    30,    6
327,    0,    1,    0,    28.5,    6
328,    0,    1,    0,    27,    6
329,    0,    1,    0,    25.5,    6
330,    0,    1,    0,    24,    6
331,    0,    1,    0,    22.5,    6
332,    0,    1,    0,    21,    6
333,    0,    1,    0,    19.5,    6
334,    0,    1,    0,    18,    6
335,    0,    1,    0,    16.5,    6
336,    0,    1,    0,    15,    6
337,    0,    1,    0,    13.5,    6
338,    0,    1,    0,    12,    6
339,    0,    1,    0,    10.5,    6
340,    0,    1,    0,    9,    6
341,    0,    1,    0,    7.5,    6
342,    0,    1,    0,    6,    6
343,    0,    1,    0,    4.5,    6
344,    0,    3,    0,    3,    6
345,    0,    4,    0,    1.5,    6
346,    0,    6,    0,    0,    6
347,    1,    7,    0,    30,    6
348,    1,    5,    2,    28.5,    6
349,    1,    1,    4,    27,    6
350,    1,    0,    5,    25.5,    6
351,    1,    0,    5,    24,    6
352,    1,    0,    13,    22.5,    6
353,    1,    1,    20,    21,    6
354,    1,    1,    26,    19.5,    6
355,    1,    1,    32,    18,    6
356,    1,    2,    37,    16.5,    6
357,    1,    1,    41,    15,    6
358,    1,    0,    44,    13.5,    6
359,    1,    0,    46,    12,    6
360,    1,    0,    48,    10.5,    6
361,    1,    1,    49,    9,    6
362,    1,    2,    49,    7.5,    6
363,    1,    1,    57,    6,    6
364,    1,    0,    64,    4.5,    6
365,    1,    1,    70,    3,    6
366,    1,    1,    76,    1.5,    6
367,    1,    0,    81,    0,    6
368,    0,    0,    85,    30,    6
369,    0,    0,    0,    28.5,    6
370,    0,    0,    0,    27,    6
371,    0,    0,    0,    25.5,    6
372,    0,    0,    0,    24,    6
373,    0,    0,    0,    22.5,    6
374,    0,    0,    0,    21,    6
375,    0,    0,    0,    19.5,    6
376,    0,    0,    0,    18,    6
377,    0,    0,    0,    16.5,    6
378,    0,    0,    0,    15,    6
379,    0,    0,    0,    13.5,    6
380,    0,    0,    0,    12,    6
```

```
    381,        0,        0,        0,     10.5,        6
    382,        0,        0,        0,        9,        6
    383,        0,        0,        0,      7.5,        6
    384,        0,        0,        0,        6,        6
    385,        0,        2,        0,      4.5,        6
    386,        0,        4,        0,        3,        6
    387,        0,        5,        0,      1.5,        6
    388,        0,        5,        0,        0,        6
    389,        1,        6,        0,       30,        6
    390,        1,        4,        4,     28.5,        6
    391,        1,        1,        7,       27,        6
    392,        1,        0,        9,     25.5,        6
    393,        1,        0,       11,       24,        6
    394,        1,        0,       12,     22.5,        6
    395,        1,        0,       12,       21,        6
    396,        1,        1,       20,     19.5,        6
    397,        1,        2,       27,       18,        6
    398,        1,        1,       33,     16.5,        6
    399,        1,        0,       39,       15,        6
    400,        1,        0,       44,     13.5,        6
    401,        1,        0,       48,       12,        6
    402,        1,        1,       51,     10.5,        6
    403,        1,        2,       53,        9,        6
    404,        1,        1,       55,      7.5,        6
    405,        1,        0,       56,        6,        6
    406,        1,        0,       56,      4.5,        6
    407,        1,        1,       64,        3,        6
    408,        1,        1,       71,      1.5,        6
    409,        1,        0,       77,        0,        6
    410,        0,        0,       83,       30,        6
    411,        0,        0,        0,     28.5,        6
    412,        0,        0,        0,       27,        6
    413,        0,        0,        0,     25.5,        6
    414,        0,        0,        0,       24,        6
    415,        0,        0,        0,     22.5,        6
    416,        0,        0,        0,       21,        6
    417,        0,        0,        0,     19.5,        6
    418,        0,        0,        0,       18,        6
    419,        0,        0,        0,     16.5,        6
    420,        0,        0,        0,       15,        6
    421,        0,        0,        0,     13.5,        6
    422,        0,        0,        0,       12,        6
    423,        0,        0,        0,     10.5,        6
    424,        0,        0,        0,        9,        6
    425,        0,        0,        0,      7.5,        6
    426,        0,        1,        0,        6,        6
    427,        0,        3,        0,      4.5,        6
    428,        0,        5,        0,        3,        6
    429,        0,        6,        0,      1.5,        6
    430,        0,        6,        0,        0,        6
    431,        1,        7,        0,       30,        6
    432,        1,        5,        6,     28.5,        6
    433,        1,        1,       11,       27,        6
    434,        1,        0,       15,     25.5,        6
    435,        1,        0,       18,       24,        6
    436,        1,        0,       20,     22.5,        6
    437,        1,        0,       22,       21,        6
    438,        1,        1,       23,     19.5,        6
    439,        1,        2,       23,       18,        6
    440,        1,        1,       31,     16.5,        6
    441,        1,        0,       38,       15,        6
    442,        1,        0,       44,     13.5,        6
    443,        1,        0,       50,       12,        6
    444,        1,        1,       55,     10.5,        6
    445,        1,        2,       59,        9,        6
    446,        1,        1,       62,      7.5,        6
```

```
447,    1,    0,    64,    6,    6
448,    1,    1,    66,    4.5,    6
449,    1,    2,    67,    3,    6
450,    1,    1,    67,    1.5,    6
451,    1,    0,    75,    0,    6
452,    0,    0,    82,    30,    6
453,    0,    0,    0,    28.5,    6
454,    0,    0,    0,    27,    6
455,    0,    0,    0,    25.5,    6
456,    0,    0,    0,    24,    6
457,    0,    0,    0,    22.5,    6
458,    0,    0,    0,    21,    6
459,    0,    0,    0,    19.5,    6
460,    0,    0,    0,    18,    6
461,    0,    0,    0,    16.5,    6
462,    0,    0,    0,    15,    6
463,    0,    0,    0,    13.5,    6
464,    0,    0,    0,    12,    6
465,    0,    0,    0,    10.5,    6
466,    0,    0,    0,    9,    6
467,    0,    0,    0,    7.5,    6
468,    0,    0,    0,    6,    6
469,    0,    2,    0,    4.5,    6
470,    0,    4,    0,    3,    6
471,    0,    5,    0,    1.5,    6
472,    0,    5,    0,    0,    6
473,    1,    6,    0,    30,    6
474,    1,    4,    7,    28.5,    6
475,    1,    1,    13,    27,    6
476,    1,    0,    19,    25.5,    6
477,    1,    0,    24,    24,    6
478,    1,    0,    28,    22.5,    6
479,    1,    0,    31,    21,    6
480,    1,    1,    33,    19.5,    6
481,    1,    2,    35,    18,    6
482,    1,    1,    36,    16.5,    6
483,    1,    0,    36,    15,    6
484,    1,    0,    44,    13.5,    6
485,    1,    0,    51,    12,    6
486,    1,    1,    57,    10.5,    6
487,    1,    2,    63,    9,    6
488,    1,    1,    68,    7.5,    6
489,    1,    0,    72,    6,    6
490,    1,    1,    75,    4.5,    6
491,    1,    2,    77,    3,    6
492,    1,    1,    79,    1.5,    6
493,    1,    0,    80,    0,    6
494,    0,    0,    80,    30,    6
495,    0,    0,    0,    28.5,    6
496,    0,    0,    0,    27,    6
497,    0,    0,    0,    25.5,    6
498,    0,    0,    0,    24,    6
499,    0,    0,    0,    22.5,    6
500,    0,    0,    0,    21,    6
501,    0,    0,    0,    19.5,    6
502,    0,    0,    0,    18,    6
503,    0,    0,    0,    16.5,    6
504,    0,    0,    0,    15,    6
505,    0,    0,    0,    13.5,    6
506,    0,    0,    0,    12,    6
507,    0,    0,    0,    10.5,    6
508,    0,    0,    0,    9,    6
509,    0,    0,    0,    7.5,    6
510,    0,    1,    0,    6,    6
511,    0,    3,    0,    4.5,    6
512,    0,    5,    0,    3,    6
```

```
513,    0,    6,    0,    1.5,    6
514,    0,    6,    0,    0,    6
515,    1,    7,    0,    30,    6
516,    1,    5,    0,    28.5,    6
517,    1,    1,    8,    27,    6
518,    1,    0,    15,    25.5,    6
519,    1,    0,    21,    24,    6
520,    1,    0,    27,    22.5,    6
521,    1,    0,    32,    21,    6
522,    1,    1,    36,    19.5,    6
523,    1,    2,    39,    18,    6
524,    1,    1,    41,    16.5,    6
525,    1,    0,    43,    15,    6
526,    1,    0,    44,    13.5,    6
527,    1,    0,    44,    12,    6
528,    1,    1,    52,    10.5,    6
529,    1,    2,    59,    9,    6
530,    1,    1,    65,    7.5,    6
531,    1,    0,    71,    6,    6
532,    1,    1,    76,    4.5,    6
533,    1,    1,    80,    3,    6
534,    1,    0,    83,    1.5,    6
535,    1,    0,    85,    0,    6
536,    0,    0,    87,    30,    6
537,    0,    0,    0,    28.5,    6
538,    0,    0,    0,    27,    6
539,    0,    0,    0,    25.5,    6
540,    0,    0,    0,    24,    6
541,    0,    0,    0,    22.5,    6
542,    0,    0,    0,    21,    6
543,    0,    0,    0,    19.5,    6
544,    0,    0,    0,    18,    6
545,    0,    0,    0,    16.5,    6
546,    0,    0,    0,    15,    6
547,    0,    0,    0,    13.5,    6
548,    0,    0,    0,    12,    6
549,    0,    0,    0,    10.5,    6
550,    0,    0,    0,    9,    6
551,    0,    0,    0,    7.5,    6
552,    0,    1,    0,    6,    6
553,    0,    3,    0,    4.5,    6
554,    0,    5,    0,    3,    6
555,    0,    5,    0,    1.5,    6
556,    0,    5,    0,    0,    6
557,    1,    6,    0,    30,    6
558,    1,    4,    2,    28.5,    6
559,    1,    1,    3,    27,    6
560,    1,    0,    3,    25.5,    6
561,    1,    0,    11,    24,    6
562,    1,    0,    18,    22.5,    6
563,    1,    1,    24,    21,    6
564,    1,    2,    30,    19.5,    6
565,    1,    1,    35,    18,    6
566,    1,    0,    39,    16.5,    6
567,    1,    0,    42,    15,    6
568,    1,    0,    44,    13.5,    6
569,    1,    1,    46,    12,    6
570,    1,    2,    47,    10.5,    6
571,    1,    1,    47,    9,    6
572,    1,    0,    55,    7.5,    6
573,    1,    0,    62,    6,    6
574,    1,    1,    68,    4.5,    6
575,    1,    1,    74,    3,    6
576,    1,    0,    79,    1.5,    6
577,    1,    0,    83,    0,    6
578,    0,    0,    86,    30,    6
```

```
579,    0,    0,    0,    28.5,    6
580,    0,    0,    0,    27,    6
581,    0,    0,    0,    25.5,    6
582,    0,    0,    0,    24,    6
583,    0,    0,    0,    22.5,    6
584,    0,    0,    0,    21,    6
585,    0,    0,    0,    19.5,    6
586,    0,    0,    0,    18,    6
587,    0,    0,    0,    16.5,    6
588,    0,    0,    0,    15,    6
589,    0,    0,    0,    13.5,    6
590,    0,    0,    0,    12,    6
591,    0,    0,    0,    10.5,    6
592,    0,    0,    0,    9,    6
593,    0,    2,    0,    7.5,    6
594,    0,    3,    0,    6,    6
595,    0,    5,    0,    4.5,    6
596,    0,    7,    0,    3,    6
597,    0,    7,    0,    1.5,    6
```